MITSUBISHI ELECTRIC

三菱電機産業用ロボット

CR750/CR751/CR760 シリーズコントローラ 取扱説明書

機能と操作の詳細解説

本書は iQ Platform 対応 CR750-Q/CR751-Q/CR760-Q コントローラと、スタンドアロンタイプの CR750-D/CR751-D/CR760-D コントローラのエラー一覧を記載したものです。

MELFA
BFP-A8868-X

⚠ 安全上のご注意

ロボットのご使用前には、必ず以下の注意事項および別冊の「安全マニュアル」を熟読し、必要な処置を講じていただくようお願いします。

A. 労働安全衛生規則（第 36 条、104 条、150 条、151 条）に基づく注意事項のポイントを示します。

⚠注意　教示作業は安全のための特別な教育を受けた作業者によりおこなってください。
（動力源を遮断しない保守作業も同様）
→安全教育の実施

⚠注意　教示作業はロボットの操作の方法および手順、異常時や再起動時の処置等に関する作業規程を作成し、これに従っておこなってください。
（動力を遮断しない保守作業も同様）
→作業規程の作成

⚠警告　教示作業は直ちに運転を停止できる装置を設けておこなってください。
（動力を遮断しない保守作業も同様）
→非常停止スイッチの設定

⚠注意　教示作業中は教示作業中である旨を始動スイッチなどに表示してください。
（動力を遮断しない保守作業も同様）
→教示作業中の表示

⚠危険　運転中は柵または囲いを設けて作業者とロボットの接触を防止してください。
→安全柵の設置

⚠注意　運転開始は関係者への一定の合図の方法を定め、これに従っておこなってください。
→運転開始の合図

⚠注意　保守作業は原則として動力を遮断しておこない、保守作業中である旨を起動スイッチなどに表示してください。
→保守作業中の表示

⚠注意　作業開始前にはロボットや非常停止スイッチ、関連装置などを点検し異常のないことを確認してください。
→作業開始前の点検

B. 別冊の「安全マニュアル」に示す注意事項のポイントを示します。
詳しくは「安全マニュアル」の本文をお読みください。

⚠危険　複数の制御機器(GOT、シーケンサ、押しボタンスイッチ)によりロボットの自動運転を行う場合、各機器の操作権などのインタロックをお客様にて設計してください。

⚠注意　ロボットは仕様範囲内の環境でお使いください。
それ以外の場合には信頼性の低下や故障の原因となります。
(温度、湿度、雰囲気、ノイズ環境等)

⚠注意　ロボットの運搬はロボットを指定の運搬姿勢にしておこなってください。
指定以外の姿勢で運搬すると落下による人身事故や故障の原因となります。

⚠注意　ロボットはしっかりとした台に据え付けてお使いください。
不安定な姿勢の場合には位置ずれや振動発生の原因となります。

⚠注意　ケーブルはノイズ源からできるだけ離して配線してください。
接近させた場合は位置ずれや誤動作の原因となります。

⚠注意　コネクタに無理な力を加えたりケーブルを過度に屈曲させないでください。
接触不良や断線の原因となります。

⚠注意　ハンドを含めたワーク質量は定格負荷および許容トルクを超えないようにしてください。
これを超えた場合にはアラーム発生や故障の原因となります。

⚠警告　ハンドや工具の取付およびワークの把持はしっかりとおこなってください。
そうでない場合には運転中の物体の飛散や放出により人身事故や物損の原因となります。

⚠警告　ロボットおよびコントローラのアースは確実に接地してください。
そうでない場合にはノイズによる誤動作や万一の場合、感電事故の原因となります。

⚠注意　ロボットの動作中は運転状態を表示するようにしてください。
表示がない場合には誤ってロボットに接近したり誤操作の原因となります。

⚠警告　ロボットの動作範囲内で教示作業をおこなう場合、必ずロボットの制御の優先権を確保してからおこなってください。そうでない場合、外部からの指令によりロボットを始動することができ、人身事故や物損の原因となります。

⚠注意　ジョグ速度はなるべく低速でおこない、ロボットから目を離さないでください。そうでない場合はワークや周辺装置との干渉の原因となります。

⚠注意　プログラム編集後の自動運転前には必ずステップ運転で動作を確認してください。そうでない場合はプログラムミス等により周辺装置との干渉の原因となります。

⚠注意　自動運転中に安全柵の出入口の扉を開けようとした場合にはロックされているか自動的にロボットが停止状態になるようにしてください。そうでない場合には人身事故の原因となります。

⚠注意　独自の判断に基づく改造や指定外の保守部品の使用はおこなわないでください。そうでない場合には故障や不具合の原因となります。

⚠警告　ロボットのアームを外部から手で動かす場合は開口部に手や指を入れないでください。姿勢によっては手や指をはさまれる場合があります。

⚠注意　ロボットの停止または非常停止をコントローラの主電源を OFF することでおこなわないでください。
自動運転中に、コントローラの主電源が OFF された場合、ロボットの精度に悪影響を及ぼす場合があります。また、アームの落下や惰走によって周辺装置等と干渉する場合があります。

⚠注意　プログラムやパラメータ等のコントローラの内部情報を書換えている時にコントローラの主電源を OFF しないでください。
自動運転中やプログラム・パラメータの書き込み中に、コントローラの主電源が OFF された場合、コントローラの内部情報が破壊される恐れがあります。

⚠危険　本製品の GOT 直結機能をお使いいただく場合、ハンディ GOT を接続しないでください。ハンディ GOT は、操作権の有効 / 無効にかかわらず、ロボットを自動運転させることができるため、物損や人身事故につながる恐れがあります。

⚠危険　CR750-Q/CR751-Q/CR760-Q で、iQ Platform 対応製品をお使いいただく場合、シーケンサにハンディ GOT を接続しないでください。ハンディ GOT は操作権の有効 / 無効にかかわらずロボットを自動運転させることができるため、物損や人身事故につながる恐れがあります。

⚠危険　マルチ CPU システムやサーボアンプの電源が投入されているときに、SSCNET III ケーブルを取りはずさないでください。モーション CPU やサーボアンプの SSCNET III コネクタおよび SSCNET III ケーブルの先端から発せられる光を直視しないでください。光が目に入ると、目に違和感を感じる恐れがあります。(SSCNET III の光源は、JIS C 6802、IEC 60825-1 に規定されているクラス 1 に相当します。)

⚠危険　コントローラの電源が投入されているときに、SSCNET III ケーブルを取りはずさないでください。SSCNET III コネクタおよび SSCNET III ケーブルの先端から発せられる光を直視しないでください。光が目に入ると、目に違和感を感じる恐れがあります。
(SSCNET III の光源は、JIS C 6802、IEC 60825-1 に規定されているクラス 1 に相当します。)

⚠危険　SSCNET III ケーブルを取りはずした後は、SSCNET III コネクタにキャップを取付けないと、ゴミやほこりの付着により、特性が劣化し、誤動作する恐れがあります。

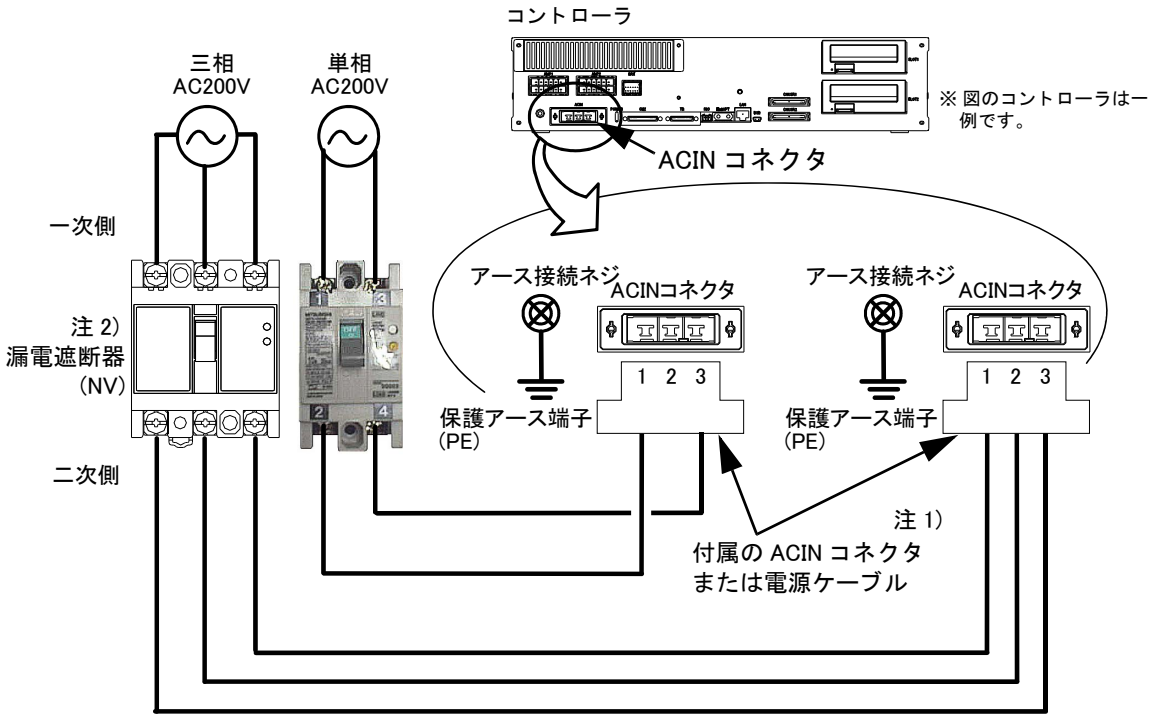
⚠注意　配線間違いがないよう十分にご注意ください。仕様以外の接続をした場合、非常停止が解除されないなどの誤動作の原因となります。
誤動作を防ぐために、配線完了後、ドライブユニット操作パネル非常停止、ティーチングボックス非常停止、お客様非常停止、ドアスイッチ等の各種機能が正常に動作することを必ず確認してください。

⚠注意　コントローラの USB を市販の機器（ノートパソコン、パソコン、LAN 用ハブなど）と接続してご使用する場合は、弊社機器との相性や温度・ノイズなどの FA 環境に適さないものもあります。
ご使用される場合は、EMI 対策（Electro-Magnetic Interference）やフェライトコアの追加など、別途対策が必要なこともありますのでお客様にて十分動作確認をお願いします。
なお、市販機器との接続における動作保障・保守は弊社でおこなうことができません。

*CR751-D または CR751-Q コントローラ

基本構成の注意事項を示します。

⚠注意　漏電保護のため、CR751-D コントローラまたは CR751-Q ドライブユニットの一次側供給電源には漏電遮断器を設置してください。そうでない場合、感電事故の原因となります。



注 1) 付属の ACIN コネクタ接続は、圧着カシメを推奨します。(はんだ付けでも可能)
推奨カシメ工具 : 234171-1(Tyco Electronics)
注 2) 漏電遮断器はお客様ご準備品です。必ず端子カバーをご使用ください。
推奨 : 単相用..... NV30FAU-2P-10A-AC100-240V-30mA(端子カバー : TCS-05FA2)
三相用..... NV30FAU-3P-10A-AC100-240V-30mA(端子カバー : TCS-05FA3)

1) 漏電遮断器 (端子カバー付)、一次電源接続用ケーブル (AWG #14(2mm^2) 以上)、一次電源接地用ケーブル (AWG #12(3.5mm^2) 以上) をご準備ください。
二次電源接続用ケーブルは、製品の仕様に合わせて単相または三相用の電源ケーブル (ACIN コネクタ付) を同梱しておりますが、同梱の ACIN コネクタと ACIN ターミナルを使用して、お客様の環境に合わせてケーブルを製作される場合は、二次電源接続用ケーブル (AWG #14(2mm^2) 以上) をご準備ください。
2) 一次電源が、電源仕様と合っていることを確認してください。
3) 一次電源が切られていることと、漏電遮断器の電源スイッチが OFF になっていることを確認してください。
4) 二次電源接続用ケーブルを接続します。
a) 同梱の電源ケーブル (ACIN コネクタ付) を使用する場合
漏電遮断器の二次側から、上図のように接続してください。
b) 同梱の ACIN コネクタと ACIN ターミナルを使用して電源ケーブルを製作する場合
ACIN ターミナルと二次電源接続用ケーブル (お客様ご準備) を接続し、ACIN ターミナルを ACIN コネクタの下記ピン番号へ挿入します。ACIN ターミナルの接続は、圧着カシメを推奨します。
単相用 : 1、3
三相用 : 1、2、3
漏電遮断器の二次側から、上図のように接続してください。
5) 電源ケーブルの ACIN コネクタをコントローラ前面の ACIN コネクタに接続します。
6) 一次電源接地用ケーブルをコントローラ筐体の保護アース端子 (M4 ネジ) に接続してください。
7) 一次電源接続用ケーブルを漏電遮断器の一次側端子に接続してください。

改定履歴

印刷日付	仕様書番号	改定内容
2011-12-20	BFP-A8868	・新規作成。
2012-01-16	BFP-A8868-A	・RH-6FH シリーズのブレーキ解除を、断続的ブレーキ解除に変更。
2012-03-14	BFP-A8868-B	・干渉回避 RT ToolBox2 利用時の回転角設定に関する注記を削除。 原点未設定時の注記を追加。 ・M_SetAdl 命令を追加。
2012-04-06	BFP-A8868-C	・電動ハンド型名別のパラメータ設定値、注記、補足説明を追加。 ・衝突検知レベル設定用サンプルプログラムを追加した(J_ColMxl) ・中部支社の所在地を変更。 （元：〒450-8522　名古屋市中村区名駅 3-28-12(大名古屋ビル)))
2012-07-26	BFP-A8868-D	・ハンド、ワーク条件パラメータに注記を追加。
2012-09-03	BFP-A8868-E	・干渉回避機能のチェック用モデルに円筒モデルを追加。 ・RV-F シリーズのパラメータ：COL の初期値と、命令語：Cmp G の最低値を追加。 ・RV-F シリーズの命令語：JRC の優先軸を追加。
2012-10-02	BFP-A8868-F	・干渉回避パラメータ：CAVSCA の出荷時設定値欄のパラメータ名称を訂正（誤記） ・衝突検知パラメータ：COL の出荷時設定値に RV-F シリーズを追加（表 5-22）
2012-10-29	BFP-A8868-G	・商標登録に関する記載を追加 ・多機能電動ハンドの新機能（テーブルデータに基づく動作など）を追加
2012-11-12	BFP-A8868-H	・パラメータ OLTMX 初期値を修正。
2013-01-09	BFP-A8868-J	・M_Uar と J_ColMxl の文例を訂正（誤記） ・パラメータ MEGDIR、MELTEXS の説明を追加。 ・垂直多関節型ロボットの構造フラグを追加
2013-02-15	BFP-A8868-K	・T/B の多機能電動ハンド操作画面を修正(ファンクションに "TBL 指定 " と " 動作 " を追加) ・RV-13F シリーズの説明を追加。(ブレーキ解除操作)
2013-07-18	BFP-A8868-M	・MELFA テレホンセンターの FAX 受付時間を修正。 ・CmpG のゲイン値の最小値を修正。(FH-F シリーズ、Cmp Jnt 時)
2013-09-20	BFP-A8868-N	・RV-4FJL の説明を追加。 ・パラメータ J1OFFSET、CMPJCLL の説明を追加。 ・RH-3FHR の説明を追加。
2014-01-06	BFP-A8868-P	・ロボット (システム) 状態変数の記述を修正。(C_Mecha、C_Prg) ・機種リセット操作追加。 ・SpdOpt 命令、SPDOPT パラメータ、円筒リミット機能 (MECAR パラメータ) を追加。 ・Def Act、Return の誤記訂正 (参照先) ・出力信号リセットパターンの説明に注記追加。 ・プログラム選択権設定の説明を修正。 ・『図 4-3：ロボットの座標系』に、原点位置を追記。 ・GetM(制御権の獲得) を必要とする命令語リストを追加。 ・EMGERR の出力条件となるエラーを追記。 ・HANDENA パラメータの注意を修正。 ・オーバーライド変更操作権に関するパラメータ (OVRDTB、OVRDENA) へ説明を追加。 ・『表 6-1：入出力信号全体マップ』内のプログラム例を修正。 (旧：If M_In(900) Then M_Out(900)=1) ・IF…Then…Else…EndIf 命令に解説 (9) を追加。 ・ROMDRV、CTN パラメータに注記を追加。(高速 RAM モードとコンティニュー機能の併用不可) ・NETIP パラメータに F-Q シリーズの出荷時設定値を追加。 ・AUTOENA パラメータの機能の説明内のエラーを L5010 に訂正。 ・プログラム容量において、誤解を招く恐れのある記述を削除。 ・Mvs に解説 (4) ～ (6) を追加。 ・PRGOUT、LINEOUT、OVRDOUT、ERROUT パラメータの、『表 6-6：専用入出力一覧』における工場出荷信号番号 (出力) を訂正。
2014-03-31	BFP-A8868-R	・スプライン補間機能を追加。 ・Ex-T 制御機能を追加。 ・OPTOVC パラメータを追加。 ・『表 5-8：ロボットの形式に対するツール変換データの有効な軸成分』内、RV-4FJL の X、Y 軸成分を△に修正。 ・パラメータを追加 (WTHFUNC、MVTERM、ESCMODE、PRGMODE、SVDATA) ・ロボット (システム) 状態変数を追加。(M_LdFMax) ・自由平面リミットの説明を補足。 ・WthIf 命令の説明を補足。 ・テレホンセンターの土・日・祝日電話受付時間を追記。
2014-04-23	BFP-A8868-S	・状態変数 M_In32 と M_Out32 の文例の誤記を訂正。

印刷日付	仕様書番号	改定内容
2014-08-19	BFP-A8868-T	・誤記修正 ・"3.9 解除できないエラーの一時的エラーリセット操作 " に R56/57TB の説明を追加。 ・専用入出力 (DOORSTS1、DOORSTS2、DOORSTS) を追加。 ・ロボット (システム) 状態変数を追加。(M_ErCode、M_DIn32、M_DOut32) ・EHServo On/OFF 命令を追加。 ・Error 命令、Act 命令の説明を修正。 ・Fine P 命令に例を追加。 ・XClr 命令の説明を修正。 ・ACos、ASin 関数を追加。
2014-12-24	BFP-A8868-U	・パラメータ JOGMENO を追加。 ・M_OPOvrd への設定の説明を追加。 ・三菱電機産業用ロボット保守サービスネットワークページの情報を更新。 (中四国支社 : 機電営業課をフィールドサービス課に変更。北日本支社 : 住所を更新。(旧 : 〒 983-0005 仙台市宮城野区福室字明神西 31)。) ・関西支社の住所、電話番号を更新。(旧 : 〒 530-8206 大阪市北区堂島 2-2-2 近鉄堂島ビル。電話番号 06-6347-2821) ・"5.11 電源 ON でのプログラムの自動実行方法 " のパラメータ ALWENA の設定値を 7 から 1 に修正し、サンプルプログラムに Step17 を追加。 ・誤記修正。 CallP 命令 (誤 : 速度の設定はサブプログラムでも有効) AUTOMATIC モード時、T/B で < 操作パネル > 画面を表示している間に点滅するランプ名。(誤 : [TB ENABLE] スイッチ) 表 6-7 : 専用入力信号の有効 / 無効状態 (誤 : 停止入力状態 (STOPSTS) が ON の時、SLOTINIT は機能しません) ・For ～ Next、While ～ WEnd 命令へ、Break 文の説明と文例を追加。 ・WithIf 命令での、Skip 命令使用時の解説を補足。 ・イラスト内のコーポレートロゴを変更。 ・Fine 命令の解説を追加。
2015-02-05	BFP-A8868-V	・テレホンセンターの FAX 技術相談を廃止。 ・XLoad 命令の誤記を訂正。(誤 : プログラム名は文字列変数でも指定可能)
2015-03-10	BFP-A8868-W	・パラメータ NETIP の説明を補足。 ・" 表 5-14 : 使用ポート番号 " を追加。 ・ソフトウェアバージョン R5q/S5q の機能を追加。 ティーチングボックスおよび RT ToolBox2 による機種リセット方法を追加。 ・パラメータ (DRVMODE、NVJBTOUT) を追加。 ・ロボット (システム) 状態変数を追加。(M_RCInfo)
2015-09-08	BFP-A8868-X	・Def Plt 命令、MvSpl 命令の説明を補足。 ・パラメータ NETIP の説明を補足。 ・命令語を追加。(EMvSpl、SplWrt、SplFWrt) ・ロボット (システム) 状態変数を追加。(M_ESpd、P_ECord) ・関数を追加。(SplSpd、SplPos、SplECord) ・"7.4 スプライン補間について " を改訂。 ・パラメータを追加。(CATEGORY) ・衝突検知に関する説明を補足。 ・干渉回避に関する説明を補足。 ・RV-50F シリーズ (RV-35F、RV-50F、RV-70F) の説明を追加。 ・パラメータ MVTERM に RV-2FL の説明を追加。

■はじめに

　このたびは、三菱電機産業用ロボットをお買い上げいただき、誠にありがとうございます。
本取扱説明書は、コントローラとティーチングボックス（R32TB/R33TB(オプション)）の機能と操作方法についての解説、および MELFA-BASIC V プログラム言語の機能と仕様についての解説を詳細に記載しています。
iQ Platform 対応の CR750-Q/CR751-Q/CR760-Q コントローラと、スタンドアロンタイプの CR750-D/CR751-D/CR760-D コントローラの両方に適用します。特に個別に付加された機能については "CR750-Q" のみ、及び "CR750-D" のみと記載しています。

　また、本書ではロボットプログラムの始動、停止などの操作はコントローラ前面の操作パネルにあるキー操作を基に説明しています。操作パネルを取付けていないロボットをご使用の場合、これらの操作は外部信号 (専用入出力信号) でおこないます。次頁に操作パネルのキーに対応する専用入出力信号をまとめてあります。パラメータの設定で汎用入出力信号に専用入出力信号を割り当ててご活用願います。

　操作される前に必ず本書を一読され、内容を十分にご理解された上でロボットをご活用いただきますようお願いいたします。
　本書では特殊な取り扱いについてもできるだけ記載するように努めておりますが、本書に記載されていない事項は「できない」と解釈してください。
なお、本書に記載している内容は、以下のソフトウェアバージョンを対象としています。

CR750-Q/CR751-Q/CR760-Q：	Ver. R6b 以上
CR750-D/CR751-D/CR760-D：	Ver. S6b 以上
T/B：	Ver. 1.7 以上

・本書の内容の一部、または全部を無断で転載することは禁止されています。
・本書の内容に関しては、将来予告なしに変更する場合があります。あらかじめご了承ください。
・本書の内容につきましては万全を期して作成しておりますが、万一ご不審な点や誤り、記載漏れなどお気づきの点がありましたら、お買い求めの販売店、または弊社 MELFA テレホンセンターへご連絡願います。連絡先は、665 ページの「7.6 技術相談窓口のお知らせ」に記載しています。
・本仕様書はオリジナルです。
・イーサネット、ETHERNET は、ゼロックス社の登録商標です。
・AutoCAD は、米国 Autodesk, Inc. の米国およびその他の国における登録商標です。
・その他記載されている会社名・製品名は各社の商標または登録商標です。

Copyright(C) 2011-2015 MITSUBISHI ELECTRIC CORPORATION

操作パネルを取付けていないロボットをご使用のお客様へ

ロボットプログラムの始動、停止などの操作は外部信号（専用入出力信号）でおこないます。本書では、コントローラ前面に操作パネルを取付けたロボットを基に、そのキー操作を説明しています。各キー操作に対応する専用入出力信号を、パラメータの設定で汎用入出力信号に割り当てて、信号操作によりおこなっていただきますようお願いします。
下表に本書で説明している操作パネルの各キーに対応する専用入出力信号を示しています。これを参考にして信号の割り当てをおこなってください。
パラメータの詳細は 516 ページの「6.3 専用入出力」に、各信号のタイムチャートは 527 ページの「6.5 外部信号のタイミングチャート」に、パラメータの設定方法は 85 ページの「3.14 パラメータ画面の操作」に記載していますので参照願います。

表：操作パネルの各キーに対応する専用入出力信号

操作パネルボタン、ランプ	パラメータ名称	区分	機能	工場出荷時設定値
START ボタン START ボタンランプ	START	入力	プログラムを始動します。	3,0
		出力	プログラムが運転中であることを出力します。	
STOP ボタン STOP ボタンランプ	STOP	入力	運転中のプログラムを停止します。	0,-1
		出力	プログラムが中断中であることを出力します。	
RESET ボタン RESET ボタンランプ	ERRRESET	入力	エラー状態を解除します。	2,2
		出力	エラー状態であることを出力します。	
	SLOTINIT	入力	プログラムの中断中を解除し、実行行を先頭に戻します。プログラム選択が可能な状態になります。	-1,-1
		出力	プログラム選択可状態であることを出力します。	
CHNG DISP ボタン UP/DOWN ボタン	PRGSEL	入力	本信号をオンすると IODATA の数値入力に割付けられた信号に入力されている値をプログラム番号として選択します。	-1,
		出力	なし	
	PRGOUT	入力	本信号をオンすると IODATA の数値出力に割付けられた信号に選択されているプログラム番号を出力します。	-1,-1
		出力	数値出力にプログラム番号出力中であることを示します。	
	OVRDSEL	入力	本信号をオンすると IODATA の数値入力に割付けられた信号に入力されている値をオーバライド値として設定します。	-1,
		出力	なし	
	OVRDOUT	入力	本信号をオンすると IODATA の数値出力に割付けられた信号に設定されているオーバライド値を出力します。	-1,-1
		出力	数値出力にオーバライド値出力中であることを示します。	
	LINEOUT	入力	本信号をオンすると IODATA の数値出力に割付けられた信号に現在の行番号を出力します。	-1,-1
		出力	数値出力に行番号出力中であることを示します。	
	ERROUT	入力	本信号をオンすると IODATA の数値出力に割付けられた信号にエラー番号を出力します。	-1,-1
		出力	数値出力にエラー番号出力中であることを示します。	
	IODATA	入力	上記、プログラム番号とオーバライド値をバイナリ値として読み込みます。	-1,-1, -1,-1
		出力	上記、プログラム番号、オーバライド値、行番号、およびエラー番号をバイナリ値として出力します。	
END ボタン END ボタンランプ	CYCLE	入力	サイクル停止をおこないます。	-1,-1
		出力	サイクル停止動作中であることを出力します。	
SVO.ON ボタン SVO.ON ボタンランプ	SRVON	入力	サーボ電源をオンします。	4,1
		出力	サーボ電源がオンしている状態を出力します。	
SVO.OFF ボタン SVO.OFF ボタンランプ	SRVOFF	入力	サーボ電源をオフします。	1,-1
		出力	サーボ電源オン不可の状態を出力します。（エコーバック）	

目次

ページ

1 ご使用にあたって 1-1
1.1 取扱説明書の使い方 1-1
1.1.1 各取扱説明書の内容 1-1
1.1.2 取扱説明書中の用語と記号について 1-2
1.2 安全上のご注意 1-3
1.2.1 労働安全衛生規則に基づく注意事項 1-3
1.2.2 安全マニュアルに示す注意事項 1-4

2 機能の解説 2-7
2.1 操作パネル(O/P)の機能(CR750 コントローラ) 2-7
(1) 操作パネルのボタン説明 2-7
(2) ステータスナンバー(表示パネル)の説明 2-7
(3) 機種リセット操作の説明 2-8
2.2 操作パネル(O/P)の機能(CR760 コントローラ) 2-9
(2) 機種リセット操作の説明 2-10
2.3 ティーチングボックス(T/B)の機能 2-11
2.3.1 操作権について 2-13
2.4 動作及び制御関連の機能 2-14

3 操作方法の解説 3-16
3.1 T/B のメニュー画面操作 3-16
(1) 画面ツリー 3-16
(2) 数字 / 文字の入力 3-20
(3) メニューの選択方法 3-21
3.2 ジョグ送り(概略説明) 3-22
3.2.1 ジョグ送りの種類 3-22
3.2.2 ジョグ送りの速度 3-24
3.2.3 関節ジョグ 3-25
3.2.4 直交ジョグ 3-25
3.2.5 ツールジョグ 3-26
3.2.6 三軸直交ジョグ 3-26
3.2.7 円筒ジョグ 3-27
3.2.8 ワークジョグ 3-27
3.2.9 ツールデータの切替 3-28
3.2.10 ワールド座標系の切替(ベース座標番号の指定) 3-29
3.2.11 ジョグ操作中の衝突検知 3-31
(1) ジョグ操作時の衝突検知レベル調整 3-31
3.3 ハンド開閉 3-32
3.4 ハンド整列 3-33
3.5 プログラミング 3-34
3.5.1 プログラムの作成 3-34
(1) プログラム編集画面のオープン 3-34
(2) プログラムの作成 3-35
(3) プログラム作成の終了、保存 3-37
(4) プログラムの修正方法 3-38
(5) 現在位置の教示 3-40
(6) 位置変数の削除 3-43
(7) 位置データの確認 3-44
(8) 位置データの MDI(Manual Data Input) 登録(修正) 3-45
3.6 デバッグ操作 3-46
(1) ステップ送り 3-46
(2) ステップ戻し 3-48
(3) 別スロットでのステップ送り 3-49
(4) ステップジャンプ 3-51
3.7 自動運転 3-52
3.7.1 動作速度の設定 3-52

目次

ページ

(1) O/P での操作 3-52
(2) T/B での操作 3-52
3.7.2 プログラムの選択 3-52
3.7.3 自動運転の開始 3-53
(1) O/P からの起動 3-53
(2) T/B からの起動 3-54
3.7.4 停止 3-57
(1) O/P での操作 3-57
(2) T/B での操作 3-57
3.7.5 停止からの自動運転再開 3-58
(1) O/P からの再起動 3-58
(2) T/B からの再起動 3-58
3.7.6 プログラムリセット 3-59
(1) O/P での操作 3-59
(2) T/B での操作 3-59
3.8 サーボオン / オフ 3-60
3.9 エラーリセット操作 3-61
3.10 解除できないエラーの一時的エラーリセット操作 3-61
3.11 管理・編集画面の操作 3-62
(1) プログラムの一覧表示 3-62
(2) プログラムのコピー (コピー) 3-63
(3) プログラムの名称変更 (リネーム) 3-64
(4) プログラムの削除 (削除) 3-65
(5) プログラムの保護 (プロテクト) 3-66
(6) プログラムの選択 3-67
3.12 運転画面の操作 3-68
3.12.1 実行行の表示 3-68
(1) 確認画面の選択 3-68
(2) ステップ運転 3-68
(3) ステップジャンプ 3-70
(4) 別スロットでのステップ送り 3-71
(5) 確認画面の終了 3-71
3.12.2 テスト運転画面 3-72
(1) テスト運転画面の選択 3-72
3.12.3 操作パネル画面の操作 3-73
3.13 モニタ画面の操作 3-74
(1) 入力信号モニタ 3-74
(2) 出力信号モニタ 3-76
(3) 入力レジスタモニタ 3-78
(4) 出力レジスタモニタ 3-79
(5) 変数モニタ 3-82
(6) エラー履歴 3-84
3.14 パラメータ画面の操作 3-85
3.15 原点・ブレーキ画面の操作 3-87
(1) 原点 3-87
(2) ブレーキ 3-87
3.16 設定・初期化画面の操作 3-89
(1) プログラムの初期化 3-89
(2) パラメータの初期化 3-90
(3) バッテリの初期化 3-91
(4) 稼動 3-92
(5) 時刻設定 3-93
(6) バージョン 3-93
3.17 拡張機能の操作 3-94
(1) シーケンサダイレクト 3-94
(2) ワーク座標 3-94

目次

ページ

3.18 初期設定画面の操作 3-95
(1) 表示言語の設定 3-95
(2) コントラストの設定 3-97

4 MELFA-BASIC V 4-99
4.1 MELFA-BASIC V の機能 4-99
4.1.1 ロボットの動作制御 4-100
(1) 関節補間動作 4-100
(2) 直線補間動作 4-101
(3) 円弧補間動作 4-102
(4) 連続動作 4-104
(5) 加減速時間と速度制御 4-105
(6) 目的位置への到達確認 4-107
(7) 高軌跡精度制御 4-108
(8) ハンド･ツール制御 4-109
4.1.2 パレット演算 4-110
4.1.3 プログラム制御 4-114
(1) 無条件分岐・条件分岐・待機 4-114
(2) 繰返し 4-116
(3) 割込み 4-117
(4) サブルーチン 4-118
(5) タイマ 4-119
(6) 停止 4-120
4.1.4 外部信号の入出力 4-121
(1) 信号入力 4-121
(2) 信号出力 4-121
4.1.5 通信 4-122
4.1.6 式と演算 4-123
(1) 演算子一覧 4-123
(2) 位置データの相対演算（乗算） 4-124
(3) 位置データの相対演算（加算） 4-124
4.1.7 付随節 4-125
4.2 MELFA-BASIC V と MELFA-BASIC IV の違い 4-126
4.2.1 MELFA-BASIC V について 4-126
4.2.2 MELFA-BASIC V の特長 4-126
4.2.3 MELFA-BASIC IV との比較 4-126
4.3 マルチタスクの機能 4-127
4.3.1 マルチタスクとは 4-127
4.3.2 マルチタスクの実行方法 4-128
4.3.3 各タスクスロットの運転状態 4-128
4.3.4 マルチタスクプログラム作成上の注意点 4-130
(1) タスク本数と処理時間の関係 4-130
(2) 並列実行するプログラム本数の最大値指定 4-130
(3) 外部変数でのプログラム間のデータの受け渡し方法 4-130
(4) ロボット（システム）状態変数でのプログラムの運転状態の確認 4-130
(5) ロボット動作のプログラムは基本的にタスクスロット 1 で実行 4-130
(6) 常時実行プログラムでの初期化処理 4-130
4.3.5 マルチタスクプログラム運転上の注意点 4-131
(1) マルチタスクの開始 4-131
(2) 運転状態の表示 4-131
4.3.6 マルチタスクの使用例 4-132
(1) ロボットの作業内容 4-132
(2) マルチタスク実行までの手順 4-133
4.3.7 プログラム容量について 4-134
(1) プログラム保存領域 4-134
(2) プログラム編集領域 4-134
(3) プログラム実行領域 4-134

目次

ページ

4.4 MELFA-BASIC V の詳細仕様 4-135
(1) プログラム名 4-135
(2) 命令文 4-135
(3) 変数 4-135
4.4.1 文 4-136
4.4.2 付随文 4-136
4.4.3 行 4-136
4.4.4 ステップ番号 4-136
4.4.5 ラベル 4-136
4.4.6 プログラムで使用できる文字の種類 4-137
4.4.7 特別な意味を持つ文字 4-138
(1) 大文字、小文字の区別 4-138
(2) アンダースコア (_) 4-138
(3) アポストロフィ (') 4-138
(4) アスタリスク (*) 4-138
(5) コンマ (,) 4-138
(6) ピリオド (.) 4-138
(7) スペース 4-138
4.4.8 データ型 4-139
4.4.9 定数 4-139
4.4.10 数値定数 4-139
(1) 10 進数 4-139
(2) 16 進数 4-139
(3) 2 進数 4-139
(4) 定数の型 4-139
4.4.11 文字列定数 4-139
4.4.12 位置定数 4-140
(1) 座標・姿勢・付加軸データの形式と意味 4-140
(2) 構造フラグデータの形式と意味 4-140
4.4.13 関節定数 4-141
(1) 各軸データの形式と意味 4-141
4.4.14 角度値 4-142
4.4.15 変数 4-142
4.4.16 数値変数 4-143
4.4.17 文字列変数 4-143
4.4.18 位置変数 4-143
4.4.19 関節変数 4-144
4.4.20 入出力変数 4-144
4.4.21 配列変数 4-144
4.4.22 外部変数 4-145
4.4.23 プログラム外部変数 4-145
4.4.24 ユーザ定義外部変数 4-146
4.4.25 ユーザベースプログラムの作成 4-146
4.5 ロボットの座標系解説 4-147
4.5.1 ロボットの座標系について 4-147
4.5.2 ベース変換について 4-148
4.5.3 位置データについて 4-149
4.5.4 ツール座標系 (メカニカルインタフェース座標系) について 4-150
(1) メカニカルインタフェース座標系 4-150
(2) ツール座標系 4-151
(3) ツール座標系使用の効果 4-152
4.6 ロボット (システム) 状態変数 4-154
4.6.1 論理数 4-157
4.7 関数 4-157
(1) ユ ― ザ定義関数 4-157
(2) 組込み関数 4-157
4.8 命令語の一覧 4-159

目次

ページ

(1) 動作制御関連の命令 4-159
(2) プログラム制御関連の命令 4-160
(3) 定義命令 4-161
(4) マルチタスク関連 4-161
(5) その他 4-161
4.9 演算子 4-162
4.10 演算の優先順位 4-163
4.11 プログラムの制御構造の深さ 4-163
4.12 予約語 4-163
4.13 命令の詳細説明 4-164
4.13.1 記載項目の見方 4-164
4.13.2 各命令語の説明 4-164
4.14 ロボット (システム) 状態変数の詳細解説 4-288
4.14.1 記載項目の見方 4-288
4.14.2 各ロボット状態変数 (システム状態変数) の説明 4-288
4.15 関数の詳細説明 4-367
4.15.1 記載項目の見方 4-367
4.15.2 各関数の説明 4-367

5 パラメータでの設定機能 5-408
5.1 動作パラメータ 5-408
5.2 信号パラメータ 5-419
5.2.1 マルチ CPU 入力オフセットについて (CR7xx-Q コントローラのみ) 5-423
(1) ケース (A) 5-423
(2) ケース (B) 5-424
5.3 操作パラメータ 5-425
5.4 プログラムパラメータ 5-427
5.5 通信パラメータ 5-430
5.6 標準ツール座標について 5-432
5.7 標準ベース座標について 5-434
5.8 ユーザ定義領域について 5-435
5.8.1 座標系の選択 5-436
5.8.2 領域の定義 5-437
5.8.3 対象メカの選択 5-439
5.8.4 ユーザ定義領域内での挙動指定 5-439
5.8.5 設定例 5-439
5.9 自由平面リミット 5-440
5.9.1 自由平面リミットの定義 5-440
5.9.2 自由平面リミットの座標系選択 5-441
5.10 中断時のジョグ送り後の自動復帰設定 5-442
5.11 電源 ON でのプログラムの自動実行方法 5-443
5.12 ハンドタイプについて 5-444
(1) 電磁弁のタイプと信号番号 5-444
5.13 ハンド初期状態について 5-445
(1) CR750/CR751 コントローラ 5-445
(2) CR760 コントローラ 5-446
5.14 出力信号リセットパターンについて 5-447
5.15 通信設定について (イーサネット) 5-449
5.15.1 各パラメータの詳細 5-449
(1) NETIP(コントローラの IP アドレス) 5-449
(2) NETMSK(サブネットマスク) 5-449
(3) NETPORT(ポート番号) 5-449
(4) CPRCE11 ～ 19(プロトコル) 5-450
(5) COMDEV(COM1 : ～ 8 に対応するデバイス定義) 5-450

目次

ページ

(6) NETMODE(サーバー指定) 5-450
(7) NETHSTIP(データ通信先のサーバーの IP アドレス) 5-450
(8) MXTTOUT(リアルタイム外部制御命令実行時のタイムアウト設定) 5-450
5.15.2 パラメータ設定例 1(ｻﾎﾟｰﾄｿﾌﾄｳｪｱを使う場合) 5-451
5.15.3 パラメータ設定例 2-1(ﾃﾞｰﾀﾘﾝｸ機能を使用：ｺﾝﾄﾛｰﾗがｻｰﾊﾞの場合) 5-452
5.15.4 パラメータ設定例 2-2(ﾃﾞｰﾀﾘﾝｸ機能を使用：ｺﾝﾄﾛｰﾗがｸﾗｲｱﾝﾄの場合) 5-453
5.15.5 パラメータ設定例 3(ﾘｱﾙﾀｲﾑ外部制御機能を使用) 5-454
5.15.6 接続の確認 5-455
5.15.7 Windows の ping コマンドによる接続の確認方法 5-455
5.16 ハンド、ワーク条件設定 (最適加減速設定) について 5-456
5.17 特異点近傍警告について 5-458
5.18 高速 RAM 運転機能について 5-459
5.19 暖機運転モードについて 5-461
5.20 特異点通過機能について 5-467
5.21 衝突検知機能について 5-472
(1) 機能の概要 5-472
(2) 関連パラメータ 5-473
(3) 衝突検知機能を使用するには 5-474
5.22 過負荷レベルの最適化について 5-479
5.23 パレット定義命令の多回転制限 5-480
5.24 干渉回避機能について 5-481
5.24.1 操作手順 5-482
5.24.2 機器の準備と接続 5-483
5.24.3 干渉チェック用モデルの登録 5-484
5.24.4 自由平面リミットの登録 5-492
5.24.5 付加軸の対応 5-493
5.24.6 共有メモリ拡張機能の設定 (ロボット同士の干渉チェックの場合のみ) 5-494
5.24.7 ロボット間キャリブレーション (ロボット同士の干渉チェックの場合のみ) 5-496
5.24.8 干渉回避機能の有効 / 無効設定 5-498
5.24.9 干渉回避機能の使用 5-499
5.24.10 サンプルプログラム 5-500
5.25 シーケンサ入出力ユニット直接制御 5-502
(1) 仕様 5-502
(2) 操作手順の概要 5-502
(3) パラメータの説明 5-503
(4) シーケンサの〝マルチ CPU 設定〟 5-504
(5) ロボット状態変数の説明 5-504
5.26 ロボット CPU 間直接通信 5-506
(1) 仕様 5-506
(2) 利用方法 5-506
(3) ロボット状態変数の説明 5-506
5.27 2 重系エラーの挙動選択パラメータについて 5-507

6 外部入出力の機能 6-508
6.1 種類 6-508
6.2 シーケンサリンク入出力機能 6-509
6.2.1 パラメータ設定 6-509
(1) シーケンサ CPU のパラメータ設定 6-509
(2) ロボット CPU のパラメータ設定 6-510
6.2.2 CPU 共有メモリとロボット入出力信号の対応 6-512
6.2.3 シーケンスラダーの例 6-512
6.2.4 専用入出力信号の割付 (工場出荷時設定) 6-514
6.3 専用入出力 6-516
6.4 信号の有効無効状態 6-526
6.5 外部信号のタイミングチャート 6-527

目次

ページ

6.5.1 各信号の個別タイミングチャート 6-527
6.5.2 タイミングチャート例 6-536
(1) 外部信号操作タイムチャート (その 1) 6-536
(2) 外部信号操作タイムチャート (その 2) 6-537
(3) 外部信号操作タイムチャート (その 3) 6-538
(4) 外部信号操作タイムチャート (その 4) 6-539
(5) 外部信号操作タイムチャート (その 5) 6-540
6.6 外部信号からのプログラム選択･実行方法 6-541
6.6.1 種類 6-541
6.6.2 実行方法の選択 6-541
6.6.3 関連する入出力パラメータ 6-541
6.6.4 操作手順 6-542
6.7 非常停止入力 6-544
6.7.1 非常停止入力時のロボットの挙動 6-544

7 付録 ..... 付録 -545
7.1 構造フラグについて ..... 付録 -545
7.2 多機能電動ハンドの使用方法 ..... 付録 -547
7.2.1 概要 ..... 付録 -547
(1) 機器構成 (例) ..... 付録 -547
7.2.2 仕様 ..... 付録 -548
7.2.3 制限事項 ..... 付録 -548
(1) 使用可能な機器とソフトウェアバージョン ..... 付録 -548
(2) 局番設定 ..... 付録 -548
(3) 多機能電動ハンドの駆動電源 (ロボット本体との関係) ..... 付録 -549
7.2.4 操作上の注意事項 ..... 付録 -549
(1) 原点復帰 ..... 付録 -549
(2) ALWAYS（常時実行）プログラムでの実行について ..... 付録 -551
7.2.5 操作手順 ..... 付録 -552
7.2.6 パラメータの設定 ..... 付録 -552
(1) アクチュエータタイプの設定 ..... 付録 -552
(2) 制御パラメータの設定 ..... 付録 -553
7.2.7 多機能電動ハンドの操作 ..... 付録 -554
(1) ハンド番号の切替え ..... 付録 -555
(2) 原点復帰 ..... 付録 -555
(3) ハンド開閉操作 ..... 付録 -556
(4) 動作位置教示 ..... 付録 -556
(5) 教示位置への移動 ..... 付録 -558
(6) 把持動作 ..... 付録 -559
(7) 動作速度の設定 ..... 付録 -560
(8) 力 (把持力) の設定 ..... 付録 -560
7.2.8 ロボット言語の仕様 ..... 付録 -561
(1) 命令語一覧 ..... 付録 -561
(2) 状態変数一覧 ..... 付録 -561
(3) 命令語の詳細説明 ..... 付録 -562
(4) 状態変数の詳細説明 ..... 付録 -572
7.2.9 パラメーター覧 ..... 付録 -576
7.2.10 エラー一覧 ..... 付録 -578
7.3 マルチハンドの使用方法 ..... 付録 -580
7.3.1 概要 ..... 付録 -580
(1) 機器構成 (例) ..... 付録 -580
(2) 使用可能な機器とソフトウェアバージョン ..... 付録 -580
7.3.2 操作手順 ..... 付録 -581
(1) パラメータの設定 ..... 付録 -581
(2) ハンド開閉操作 (R32TB) ..... 付録 -584
(3) ツール番号 (ツール座標パラメータ) とマルチハンドとの関係 ..... 付録 -585
(4) アラームについて ..... 付録 -586

目次

ページ

7.4 スプライン補間について ..... 付録 -587
7.4.1 概要 ..... 付録 -587
(1) 概要 ..... 付録 -587
(2) 特徴 ..... 付録 -587
(3) 必要な機器とソフトウェアバージョン ..... 付録 -588
(4) 用語 ..... 付録 -588
7.4.2 仕様 ..... 付録 -589
(1) 基本仕様 ..... 付録 -589
(2) 制限事項 ..... 付録 -590
(3) スプライン補間におけるロボットの挙動 ..... 付録 -590
(4) 経路点に関するチェック ..... 付録 -592
7.4.3 機能の説明 ..... 付録 -593
(1) 経路補正 ..... 付録 -593
(2) 動作モード ..... 付録 -596
(3) 信号出力 ..... 付録 -597
(4) 数値設定 ..... 付録 -598
(5) フレーム変換 ..... 付録 -598
7.4.4 作業手順 ..... 付録 -600
7.4.5 スプラインファイルの作成 ..... 付録 -602
(1) 新規作成 ..... 付録 -602
(2) DXF ファイルインポート機能で作成 ..... 付録 -603
(3) ロボット言語で新規作成 ..... 付録 -609
(4) 既存のスプラインファイルを開く ..... 付録 -609
(5) スプラインファイル編集画面の説明 ..... 付録 -611
(6) スプラインファイル編集時のメニューバー ..... 付録 -613
(7) 経路点データの内容 ..... 付録 -615
(8) スプラインファイルの編集 ..... 付録 -616
(9) スプラインファイルの保存 ..... 付録 -622
(10) スプラインファイルの削除 ..... 付録 -624
(11) スプライン番号の変更 ..... 付録 -624
(12) スプラインファイルのコピー ..... 付録 -625
(13) スプラインファイル管理 ..... 付録 -626
(14) インポート・エクスポート機能 ..... 付録 -627
(15) 編集の補助機能 ..... 付録 -636
(16) スプライン曲線の表示 ..... 付録 -638
(17) 編集内容の確認 ..... 付録 -638
(18) ロボットプログラム ..... 付録 -639
7.4.6 ロボットプログラムの作成 ..... 付録 -642
7.4.7 動作の確認 ..... 付録 -645
7.4.8 ロボットコントローラへ保存 ..... 付録 -645
7.4.9 調整作業 ..... 付録 -645
(1) 位置補正機能 ..... 付録 -645
(2) フレーム変換機能 ..... 付録 -647
(3) ポジションジャンプ ..... 付録 -650
(4) パラメータ SPLOPTGC ..... 付録 -651
7.5 Ex-T 制御について ..... 付録 -652
7.5.1 概要 ..... 付録 -652
(1) 特徴 ..... 付録 -652
(2) 仕様 ..... 付録 -653
(3) 必要な機器とソフトウェアバージョン ..... 付録 -653
7.5.2 Ex-T 座標の設定 ..... 付録 -653
(1) 設定方法 ..... 付録 -653
7.5.3 Ex-T ジョグ ..... 付録 -654
(1) ワークジョグにおける姿勢成分動作 ..... 付録 -655
(2) Ex-T ジョグにおける姿勢成分動作 ..... 付録 -655
(3) Ex-T ジョグの操作 ..... 付録 -656
(4) RV6 軸タイプのワークジョグ動作 ..... 付録 -657
(5) RH4 軸タイプのワークジョグ動作 ..... 付録 -659

目次

ページ

(6) RH4 軸天吊りタイプのワークジョグ動作 ..... 付録 -661
7.5.4 ロボットプログラムの作成 ..... 付録 -663
(1) Ex-T 制御関連命令・変数の一覧 ..... 付録 -663
(2) プログラミング例 ..... 付録 -663
7.6 技術相談窓口のお知らせ ..... 付録 -665
7.7 アフターサービスについて ..... 付録 -665

1 ご使用にあたって

本章では、取扱説明書の内容と使い方や、基本的な用語、安全事項を説明しています。
また、各取扱説明書中のティーチングボックス(T/B)の取扱、操作方法はR32TB(R33TB)を基に記載しております。R56TB(R57TB)など他のT/Bをご使用場合は、それぞれのT/Bに付属の取扱説明書を参照願います。

1.1 取扱説明書の使い方

1.1.1 各取扱説明書の内容

本製品に添付しておりますドキュメントの内容、目的等について以下に示します。
用途に応じてご活用ください。
なお、特殊仕様の場合にはその特殊部分を説明した分冊の取扱説明書を付属する場合があります。

安全マニュアル	ロボットに関わるすべての作業者の安全を確保するために、ロボットの取り扱い、システム設計および製作についての共通的な注意事項と安全対策について説明しています。
標準仕様書	製品の標準仕様や工場出荷特殊仕様、オプション構成、保守部品等について説明しています。 また、ロボットの導入にあたっての安全上、技術上の注意事項についても解説しています。
ロボット本体セットアップから保守まで	ロボット本体について、動作できるようにするまでの手順(開梱、運搬、設置、動作確認)と、その保守、点検について説明しています。
コントローラセットアップと基本操作から保守まで	コントローラについて、操作できるようにするまでの手順(開梱、運搬、設置、動作確認)と、プログラム作成から自動運転までの基本操作、および保守、点検について説明しています。
機能と操作の解説	各機能の解説や操作方法、プログラムで使用するMELFA-BASIC Vコマンドの解説と、外部入出力機器との接続方法、パラメータの解説など、機能・操作の詳細を解説しています。
トラブルシューティング	エラー発生時、そのエラー番号に対する原因と対策を説明しています。
付加軸機能	ロボット本体に走行軸などを付加させてご使用になる場合に使用する汎用サーボアンプの制御機能を解説しています。
トラッキング機能	コンベアトラッキングの仕様、機能、使用方法を説明しています。
拡張機能説明書	iQ Platform対応ロボットにおけるシーケンサ、またはスタンドアローンタイプロボットにおけるGOTとロボット間のメモリのデータ構成やモニタ、操作手順を詳細に解説しています。

1.1.2 取扱説明書中の用語と記号について

取扱説明書では、表 1-1 のような用語と記号を使って表現しています。
ご確認の上取扱説明書を読み進めてください。

表 1-1：取扱説明書中の記号

区分	用語／記号	意味
用語	iQ Platform	
	コントローラ	ロボット本体を制御するコントローラを示します。 ロボット CPU システム、ドライブユニットから構成された制御部の総称を示します。
	ロボット CPU ユニット または ロボット CPU	当社 MELSEC-Q シリーズのシーケンサベースユニット（Q3 □ DB）にスロットインされる、ロボット用 CPU ユニットを示します。ドライブユニットと専用ケーブルで接続されます。
	ロボット CPU システム	マルチ CPU システム。 シーケンサベースユニット、シーケンサ CPU ユニット、ロボット CPU ユニット、その他、MELSEC ユニットから構成されます。
	ドライブユニット	ロボット用サーボアンプ、安全回路などを搭載したボックスを示します。
記号	⚠ 危険	取り扱いを誤った場合、使用者が死亡、または重傷を負う危険が切迫して生じることが想定される場合の注意事項です。 ロボットを安全にお使いになるために必ずおこなってください。
	⚠ 警告	取り扱いを誤った場合、使用者が死亡、または重傷を負う可能性が想定される場合の注意事項です。 ロボットを安全にお使いになるために必ずおこなってください。
	⚠ 注意	取り扱いを誤った場合、使用者が障害を負う危険が想定される場合、または物的損害のみの発生が想定される場合の注意事項です。 ロボットを安全にお使いになるために必ずおこなってください。
	例）［ JOG ］	［ ］をつけているものは、ドライブユニット前面操作パネルのスイッチ（操作パネル付仕様）、およびティーチングボックスのキーを示しています。
	例)[RESET] ＋ [EXE] （A）　（B）	（A）キーを押したまま（B）キーを押すことを示しています。 この例では、[RESET] キーを押したまま [EXE] キーを押します。
	T/B	ティーチングボックスのことを示します。 本書では、R32TB をベースに解説しています。
	O/P	コントローラまたはドライブユニット前面の操作パネル (操作パネル付仕様) を示します。
	CR751(薄型) CR751(厚型)	CR751 コントローラには薄型 (高さ 98mm) と厚型 (高さ 174mm) の 2 種類があります。 薄型：CR751-03HD/Q、CR751-06HD/Q、CR751-12HD/Q、CR751-20HD/Q、CR751-03HRD/Q、CR751-02VD/Q、CR751-04VD/Q、CR751-04VJD/Q、CR751-07VD/Q 厚型：CR751-13VD/Q、CR751-20VD/Q、CR751-07VLD/Q ※CR751 コントローラの外形図は、別冊の「標準仕様書」を参照ください。

1.2 安全上のご注意

ロボットのご使用前には、必ず以下の注意事項および別冊の「安全マニュアル」を熟読し、必要な処置を講じていただくようお願いします。

1.2.1 労働安全衛生規則に基づく注意事項

以下に労働安全衛生規則（第 36 条、104 条、150 条、151 条）に基づく注意事項のポイントを示します。

⚠注意　教示作業は安全のための特別な教育を受けた作業者によりおこなってください。
（動力源を遮断しない保守作業も同様）
→安全教育の実施

⚠注意　教示作業はロボットの操作の方法および手順、異常時や再起動時の処置等に関する作業規程を作成し、これに従っておこなってください。
（動力を遮断しない保守作業も同様）
→作業規程の作成

⚠警告　教示作業は直ちに運転を停止できる装置を設けておこなってください。
（動力を遮断しない保守作業も同様）
→非常停止スイッチの設定

⚠注意　教示作業中は教示作業中である旨を始動スイッチなどに表示してください。
（動力を遮断しない保守作業も同様）
→教示作業中の表示

⚠危険　運転中は柵または囲いを設けて作業者とロボットの接触を防止してください。
→安全柵の設置

⚠注意　運転開始は関係者への一定の合図の方法を定め、これに従っておこなってください。
→運転開始の合図

⚠注意　保守作業は原則として動力を遮断しておこない、保守作業中である旨を起動スイッチなどに表示してください。
→保守作業中の表示

⚠注意　作業開始前にはロボットや非常停止スイッチ、関連装置などを点検し異常のないことを確認してください。
→作業開始前の点検

1.2.2 安全マニュアルに示す注意事項

以下に、別冊の「安全マニュアル」に示す注意事項のポイントを示します。詳しくは「安全マニュアル」の本文をお読みください。

⚠危険 複数の制御機器(GOT、シーケンサ、押しボタンスイッチ)によりロボットの自動運転を行う場合、各機器の操作権などのインタロックをお客様にて設計してください。

⚠注意 ロボットは仕様範囲内の環境でお使いください。
それ以外の場合には信頼性の低下や故障の原因となります。
(温度、湿度、雰囲気、ノイズ環境等)

⚠注意 ロボットの運搬はロボットを指定の運搬姿勢にしておこなってください。
指定以外の姿勢で運搬すると落下による人身事故や故障の原因となります。

⚠注意 ロボットはしっかりとした台に据え付けてお使いください。
不安定な姿勢の場合には位置ずれや振動発生の原因となります。

⚠注意 ケーブルはノイズ源からできるだけ離して配線してください。
接近させた場合は位置ずれや誤動作の原因となります。

⚠注意 コネクタに無理な力を加えたりケーブルを過度に屈曲させないでください。
接触不良や断線の原因となります。

⚠注意 ハンドを含めたワーク質量は定格負荷および許容トルクを超えないようにしてください。
これを超えた場合にはアラーム発生や故障の原因となります。

⚠警告 ハンドや工具の取付およびワークの把持はしっかりとおこなってください。
そうでない場合には運転中の物体の飛散や放出により人身事故や物損の原因となります。

⚠警告 ロボットおよびコントローラのアースは確実に接地してください。
そうでない場合にはノイズによる誤動作や万一の場合、感電事故の原因となります。

⚠注意 ロボットの動作中は運転状態を表示するようにしてください。
表示がない場合には誤ってロボットに接近したり誤操作の原因となります。

⚠警告 ロボットの動作範囲内で教示作業をおこなう場合、必ずロボットの制御の優先権を確保してからおこなってください。そうでない場合、外部からの指令によりロボットを始動することができ、人身事故や物損の原因となります。

⚠注意 ジョグ速度はなるべく低速でおこない、ロボットから目を離さないでください。そうでない場合はワークや周辺装置との干渉の原因となります。

⚠注意 プログラム編集後の自動運転前には必ずステップ運転で動作を確認してください。そうでない場合はプログラムミス等により周辺装置との干渉の原因となります。

⚠注意 自動運転中に安全柵の出入口の扉を開けようとした場合にはロックされているか自動的にロボットが停止状態になるようにしてください。そうでない場合には人身事故の原因となります。

⚠注意　独自の判断に基づく改造や指定外の保守部品の使用はおこなわないでください。そうでない場合には故障や不具合の原因となります。

⚠警告　ロボットのアームを外部から手で動かす場合は開口部に手や指を入れないでください。姿勢によっては手や指をはさまれる場合があります。

⚠注意　ロボットの停止または非常停止をコントローラの主電源を OFF することでおこなわないでください。
自動運転中に、コントローラの主電源が OFF された場合、ロボットの精度に悪影響を及ぼす場合があります。また、アームの落下や惰走によって周辺装置等と干渉する場合があります。

⚠注意　プログラムやパラメータ等のコントローラの内部情報を書換えている時にコントローラの主電源を OFF しないでください。
自動運転中やプログラム・パラメータの書き込み中に、コントローラの主電源が OFF された場合、コントローラの内部情報が破壊される恐れがあります。

⚠危険　本製品の GOT 直結機能をお使いいただく場合、ハンディ GOT を接続しないでください。ハンディ GOT は、操作権の有効 / 無効にかかわらず、ロボットを自動運転させることができるため、物損や人身事故につながる恐れがあります。

⚠危険　CR750-Q/CR751-Q/CR760-Q で、iQ Platform 対応製品をお使いいただく場合、シーケンサにハンディ GOT を接続しないでください。ハンディ GOT は操作権の有効 / 無効にかかわらずロボットを自動運転させることができるため、物損や人身事故につながる恐れがあります。

⚠危険　マルチ CPU システムやサーボアンプの電源が投入されているときに、SSCNET III ケーブルを取りはずさないでください。モーション CPU やサーボアンプの SSCNET III コネクタおよび SSCNET III ケーブルの先端から発せられる光を直視しないでください。光が目に入ると、目に違和感を感じる恐れがあります。(SSCNET III の光源は、JIS C 6802、IEC 60825-1 に規定されているクラス 1 に相当します。)

⚠危険　コントローラの電源が投入されているときに、SSCNET III ケーブルを取りはずさないでください。SSCNET III コネクタおよび SSCNET III ケーブルの先端から発せられる光を直視しないでください。光が目に入ると、目に違和感を感じる恐れがあります。
(SSCNET III の光源は、JIS C 6802、IEC 60825-1 に規定されているクラス 1 に相当します。)

⚠危険　SSCNET III ケーブルを取りはずした後は、SSCNET III コネクタにキャップを取付けないと、ゴミやほこりの付着により、特性が劣化し、誤動作する恐れがあります。

⚠注意　配線間違いがないよう十分にご注意ください。仕様以外の接続をした場合、非常停止が解除されないなどの誤動作の原因となります。
誤動作を防ぐために、配線完了後、ドライブユニット操作パネル非常停止、ティーチングボックス非常停止、お客様非常停止、ドアスイッチ等の各種機能が正常に動作することを必ず確認してください。

⚠注意

コントローラの USB を市販の機器（ノートパソコン、パソコン、LAN 用ハブなど）と接続してご使用する場合は、弊社機器との相性や温度・ノイズなどの FA 環境に適さないものもあります。
ご使用される場合は、EMI 対策（Electro-Magnetic Interference）やフェライトコアの追加など、別途対策が必要なこともありますのでお客様にて十分動作確認をお願いします。
なお、市販機器との接続における動作保障・保守は弊社でおこなうことができません。

2 機能の解説

2.1 操作パネル(O/P)の機能(CR750コントローラ)

(1) 操作パネルのボタン説明

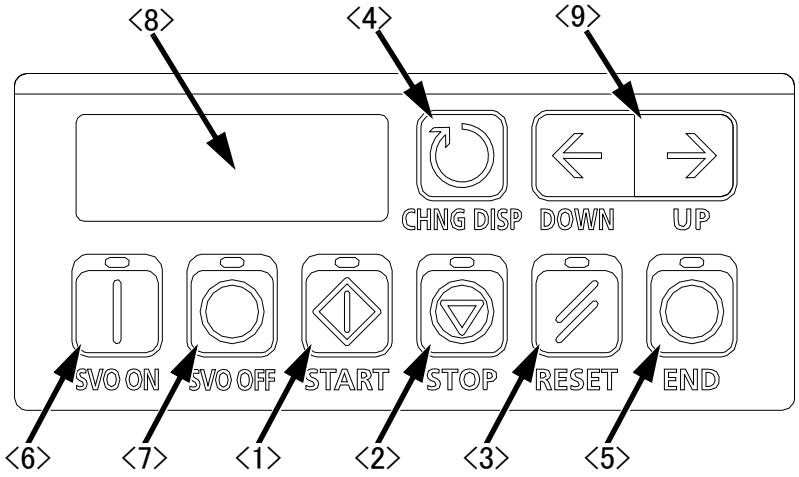


図 2-1：操作パネル各部の名称（コントローラ）

<1> スタートボタン.....プログラムを実行し、ロボットを動作させます。プログラムは連続運転です。
<2> ストップボタン.....ロボットを直ちに停止させます。サーボオフしません。
<3> リセットボタン.....エラーを解除します。また、プログラムの中断中状態を解除し、プログラムリセットします。
<4> 表示切替ボタン.....表示パネルの表示内容を「オーバーライド」→「行番号」→「プログラム番号」→「ユーザ情報」→「メーカ情報」の順に切り替えます。
<5> エンドボタン.....実行中のプログラムの最終行または End 文で停止させます。
<6>SVO.ON ボタン .....サーボ電源を入れます。（サーボオンします。）
<7>SVO.OFF ボタン.....サーボ電源を切ります。（サーボオフします。）
<8>STATUS.NUMBER（表示パネル）
アラーム番号、プログラム番号、オーバーライド値(%)などを表示します。
<9>UP/DOWN ボタン .....表示パネルの表示内容をアップ表示、ダウン表示します。

(2) ステータスナンバー（表示パネル）の説明

プログラム名にアルファベットを使った場合に、操作パネルの 7 セグの LED に表示する際の簡易表示の記号を以下に示します。

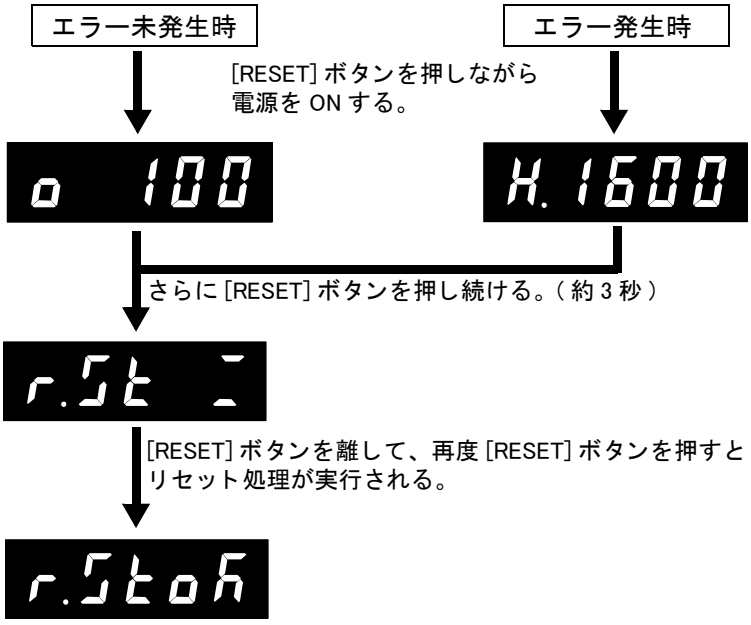
A		B		C		D		E		F		G		H		I	
J		K		L		M		N		O		P		Q		R	
S		T		U		V		W		X		Y		Z		?	

プログラム名表示では先頭に "P" がつきますので表示可能な文字は 4 文字以内です。よって、プログラム名は 4 文字以内で作成してください。

4 文字以上のプログラム名は操作パネルからは選択できません。ロボット言語の CallP 命令にてサブプログラムとして実行するプログラムの場合は 4 文字以上で作成してもかまいません。

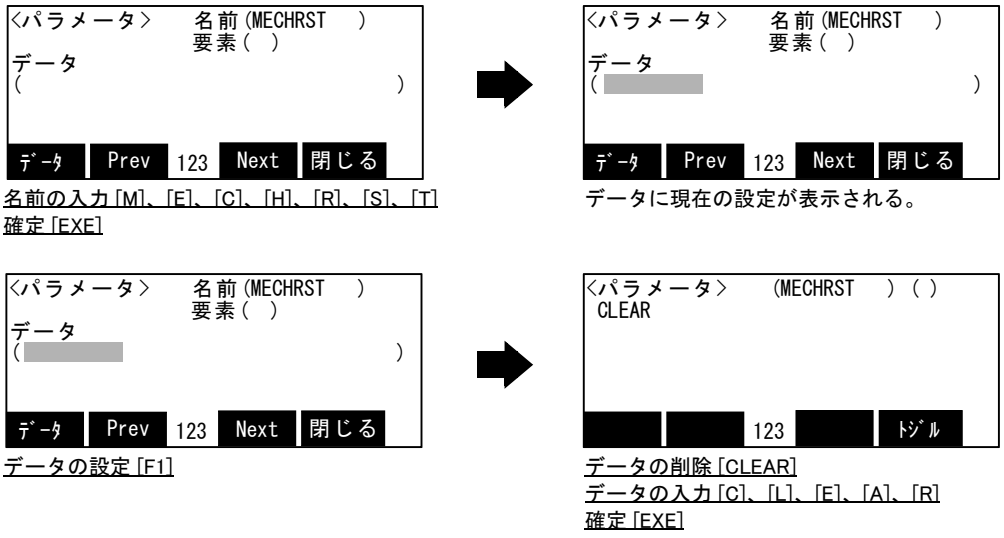
(3) 機種リセット操作の説明

バッテリの消耗などで機種情報が消えてしまうとエラー（H1600：メカ未設定　など）が発生して、操作ができなくなります。その場合は、以下の機種リセット操作をおこないますと、工場出荷時状態になり、正しい機種情報が設定できます。



コントローラのソフトウェアバージョンが R5q(F-Q シリーズ)/S5q(F-D シリーズ) 以降の場合、ティーチングボックスまたは RT ToolBox2 から機種リセット操作をおこなうことができます。
以下に、ティーチングボックスによる機種リセット操作の方法を示します。

1) パラメータ MECHRST に CLEAR を設定します。



2) コントローラの電源を再投入します。

⚠ 注意　本操作をおこないますと工場出荷時状態になり、パラメータ、プログラムと各種ログデータは消去されますので、エラー（H1600）が発生している時以外は実施しないでください。消去されてしまったこれらのデータは、RT ToolBox / RT ToolBbox2(オプション) の一括バックアップデータがあれば、リストア機能により復元することができます。

2.2 操作パネル(O/P)の機能(CR760コントローラ)

(1) 操作パネルのボタン説明

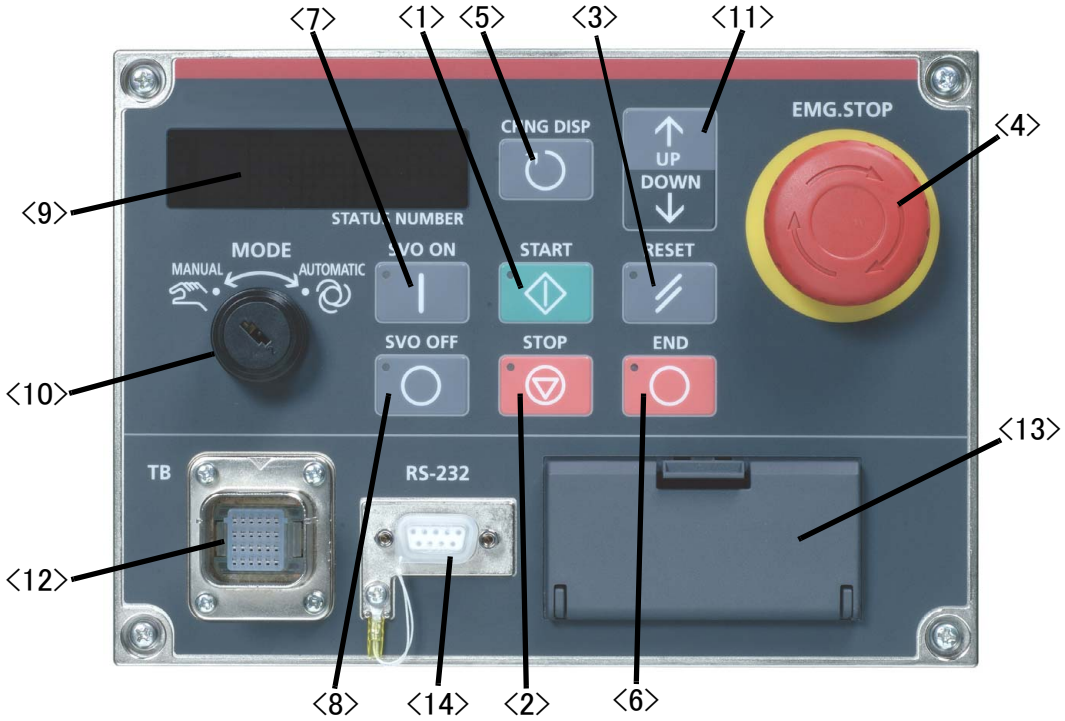
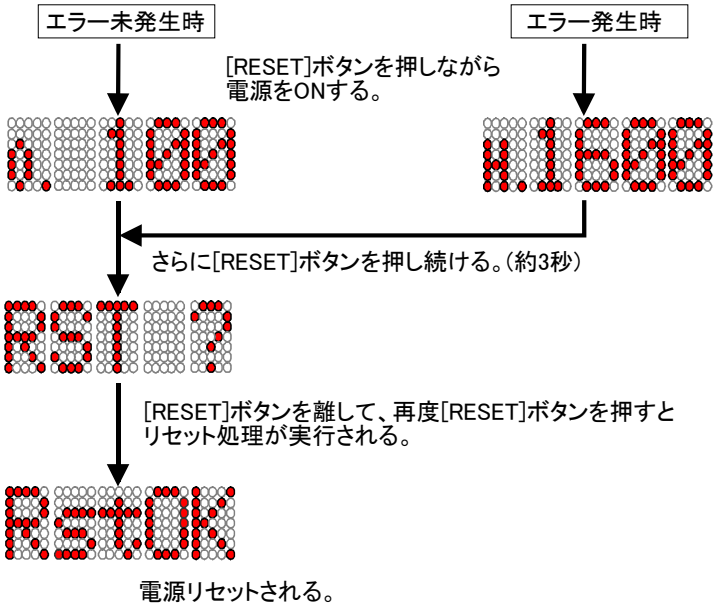


図 2-2：操作パネル各部の名称（コントローラ）

<1> スタートボタン.....プログラムを実行し、ロボットを動作させます。プログラムは連続運転です。
<2> ストップボタン.....ロボットを直ちに停止させます。サーボオフしません。
<3> リセットボタン.....エラーを解除します。また、プログラムの中断中状態を解除し、プログラムリセットします。
<4> 非常停止スイッチ.....ロボットを緊急停止します。サーボオフします。
<5> 表示切替ボタン.....表示パネルの表示内容を「オーバーライド」→「行番号」→「プログラム番号」→「ユーザ情報」→「メーカ情報」の順に切り替えます。
<6> エンドボタン.....実行中のプログラムの最終行または End 文で停止させます。
<7>SVO.ON ボタン .....サーボ電源を入れます。（サーボオンします。）
<8>SVO.OFF ボタン.....サーボ電源を切ります。（サーボオフします。）
<9>STATUS.NUMBER(表示パネル)
アラーム番号、プログラム番号、オーバーライド値 (%) などを表示します。
<10>MODE 切替スイッチ.....ロボットの操作権を切り替えます。
AUTOMATIC.....操作パネルまたは外部機器からの操作が有効となります。T/B からの操作権を必要とする操作はできません。操作パネルと外部機器との接続には操作権用のパラメータを設定する必要があります。詳細については、「13 ページの「2.3.1 操作権について」」を参照ください。
MANUAL.....T/B が有効の時、T/B からの操作のみ有効となります。外部機器や操作パネルからの操作権を必要とする操作はできません。
<11>UP/DOWN ボタン.....表示パネルの表示内容をアップ表示、ダウン表示します。
<12>T/B 接続用コネクタ.....T/B を接続する専用のコネクタです。T/B を使用しない時は、付属のダミーコネクタを接続します。
<13> インタフェースカバー....カバーを開けると USB インタフェース及びバッテリが搭載されています。CR760-Q コントローラにはありません。
<14>RS-232 コネクタ.....パソコンを接続する RS-232 仕様のコネクタです。CR760-Q コントローラにはありません。

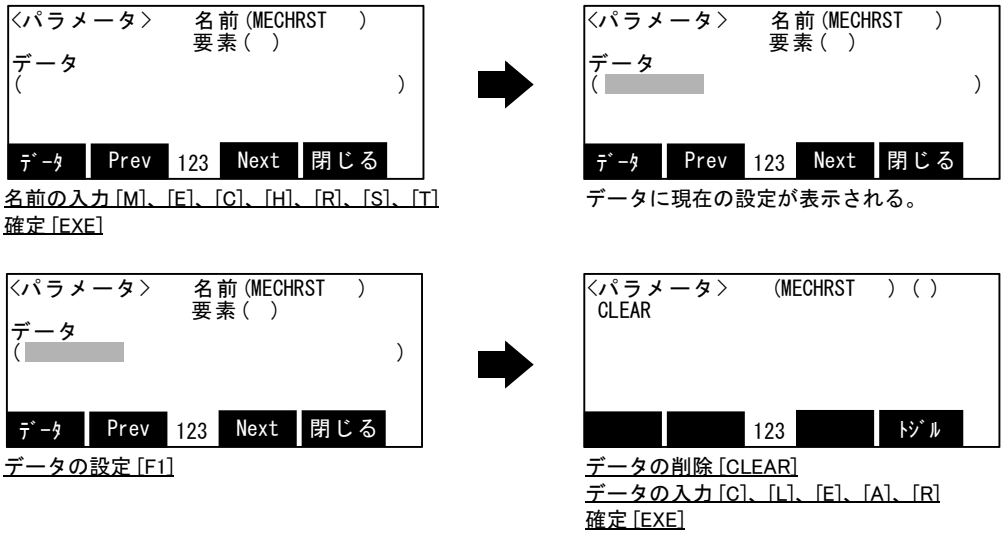
(2) 機種リセット操作の説明

バッテリの消耗などで機種情報が消えてしまうとエラー（H1600：メカ未設定　など）が発生して、操作ができなくなります。その場合は、以下の機種リセット操作をおこないますと、工場出荷時状態になり、正しい機種情報が設定できます。



ティーチングボックスまたは RT ToolBox2 から機種リセット操作をおこなうことができます。
以下に、ティーチングボックスによる機種リセット操作の方法を示します。

1) パラメータ MECHRST に CLEAR を設定します。



2) コントローラの電源を再投入します。

⚠ 注意　本操作をおこないますと工場出荷時状態になり、パラメータ、プログラムと各種ログデータは消去されますので、エラー（H1600）が発生している時以外は実施しないでください。消去されてしまったこれらのデータは、RT ToolBox / RT ToolBbox2(オプション)の一括バックアップデータがあれば、リストア機能により復元することができます。

2.3 ティーチングボックス(T/B)の機能

本章では、R32TB/R33TB(オプション) の機能を記述します。

(1) 各キーの機能

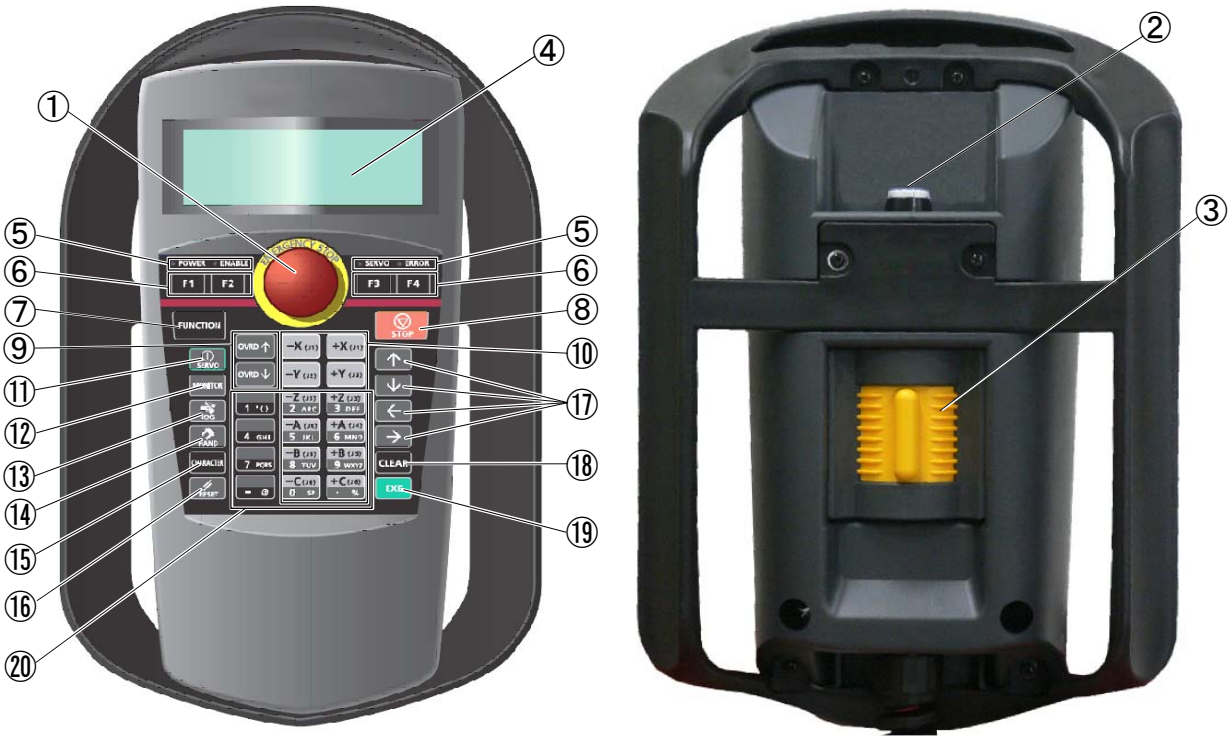


図 2-3 : T/B 外観図

① [EMG. STOP] スイッチ
非常停止用のロック機能付き押しボタンスイッチです。このスイッチを押すと、T/B の有効 / 無効状態にかかわらず、ロボットはサーボオフし、直ちに停止します。
非常停止状態を解除する場合は、スイッチを右に回すか、または手前に引いてください。(非常停止スイッチを押すとロボットはエラー状態になっています。スイッチを解除した後、エラーのリセット操作を行ってください。)

② [TB ENABLE] スイッチ
T/B によるロボットの操作を有効とするか無効とするかするかのスイッチです。このスイッチはオルタネートスイッチで、T/B が有効の時は、スイッチ内のランプが点灯します。また、前面の ENABLE ランプも点灯します。T/B を用いてロボットを操作する時は、必ず T/B 有効としてください。
T/B を有効にすると、T/B での操作に優先権が与えられて T/B での操作が可能となると共にコントローラおよび外部からの操作ができなくなります。なお、コントローラまたは外部から操作する場合には、T/B 無効状態にしてください。

③イネーブルスイッチ (3 ポジションスイッチ)
背面にある 3 ポジションのスイッチです。MANUAL モードの時、、本スイッチを離すか、または強く押すとサーボオフします。ジョグ操作やステップ実行などサーボオン状態で機能する操作は、本スイッチを軽く押さえたままで行ってください。
なお、非常停止やサーボオフ操作を行ってサーボオフしている状態では、本スイッチだけを押してもサーボオンしません。あらためてサーボオン操作を行ってください。

④表示パネル
T/B のキー操作により、プログラムの内容やロボットの状態を表示します。

⑤状態表示ランプ
T/B やロボットの状態を表示します。
[POWER] : T/B に電源が供給されているとき緑色に点灯します。
[ENABLE]: T/B が有効状態のとき緑色に点灯します。
[SERVO] : ロボットがサーボオンしているとき緑色に点灯します。
[ERROR] : ロボットがエラー状態のとき赤色に点灯します。

⑥ [F1][F2][F3][F4] キー
表示パネルのファンクション表示部に表示されている機能を実行します。
⑦ [FUNCTION] キー
1 つの操作において、[F1][F2][F3][F4] キーに割り当てる機能が 5 つ以上有るときに、このキーを押すとファンクション表示を切り換えて、[F1][F2][F3][F4] キーに割り当てる機能を変更します。
画面の最下段に白抜き文字で表示されたメニューがあります。これらは左から順に [F1][F2][F3][F4] キーに割り当てられています。対応するファンクションキーを押すことで表示されているメニューを選択することができます。
また、メニューの右端に〝⇒〟が表示されている場合は、現在の表示の他にもまだメニューがあることを示しており、[FUNCTION] キーを押すごとに表示メニューが切り替わります。
⑧ [STOP] キー
ロボットが動作中であれば、即時に減速し、ロボットの動作を停止させます。また、プログラム実行中であれば、実行を中断します。コントローラ前面の [STOP] スイッチと同じ機能です。T/B が接続されている状態であれば、[ENABLE] スイッチが押されていない場合 ([ENABLE] ランプが点灯していない場合) にも使用可能です。
⑨ [OVRD ↑][OVRD ↓] キー
ロボットの速度オーバーライドを変化させます。[OVRD ↑] キーを押すとオーバーライドが上がります。[OVRD ↓] キーを押すとオーバーライドが下がります。このキー操作によるオーバーライドの変更は、コントローラ前面のオーバーライド表示も変化させます。
⑩ [ジョグ操作] キー ([-X(J1)] ～ [+C(J6)] の 12 個のキー)
T/B がジョグモードのときは、このキーによりジョグ操作を行います。
また、T/B がハンド操作モードのときは、このキーによりハンド操作を行います。
⑪ [SERVO] キー
(3 ポジションスイッチ) を軽く握りながら、このキーを押すと、ロボットはサーボオンします。
⑫ [MONITOR] キー
このキーを押すと、モニターモードとなり、モニターメニューを表示します。モニターモードのときにこのキーを押すと、モニターモードになる前の画面に戻ります。
⑬ [JOG] キー
このキーを押すと、ジョグモードとなり、ジョグ画面を表示します。ジョグモードのときにこのキーを押すと、ジョグモードになる前の画面に戻ります。
⑭ [HAND] キー
このキーを押すと、ハンド操作モードとなり、ハンド操作画面を表示します。ハンド操作モードのときにこのキーを押すと、ハンド操作モードになる前の画面に戻ります。
また、このキーを 2 秒以上押し続けると、ツール選択画面となり、ツールデータ選択を行うモードに変わります。ツール選択モードのときにこのキーを 2 秒以上押し続けると、前の画面に戻ります。
⑮ [CHARACTER] キー
T/B が文字入力または数字入力が可能なときに、[数字 / 文字] キーの機能を数字入力と文字入力との間で切り換えます。
⑯ [RESET] キー
ロボットがエラー状態のとき、エラーを解除します。(解除できないエラーもあります)
また、このキーを押しながら [EXE] キーを押すことで、プログラムリセットを行います。
⑰ [↑][↓][←][→] キー
カーソルをそれぞれの方向に移動させます。
⑱ [CLEAR] キー
数字入力または文字入力が可能なときに、このキーを押すとカーソルのある文字を 1 文字削除します。
また、長押しすることで、カーソルがある入力領域をすべてクリアします。
⑲ [EXE] キー
入力操作を確定します。
また、ダイレクト実行のときは、このキーを押し続けている間、ロボットは動作します。
⑳ [数字 / 文字] キー
数字入力または文字入力が可能なときに、このキーを押すと数字または文字を表示します。

2.3.1 操作権について

コントローラに T/B やパソコンなど、複数の機器が接続されていても、同時にコントローラに対する操作（運転、サーボオンなどの指令）が有効な機器は 1 つに限られます。この限られた 1 つの機器が「操作権を得ている」といいます。
プログラム始動やサーボオンなどロボットを起動する、または起動できる要因となるような操作は操作権がないとできませんが、反対に、停止やサーボオフなどロボットを停止させるような操作は安全上操作権がなくても働くようになっています。

表 2-1：設定スイッチと操作権の関係　　○：操作権有り、×：操作権無し

設定スイッチ	T/B[ENABLE]	無効 注 1)		有効	
	コントローラのモード	AUTOMATIC	MANUAL	AUTOMATIC	MANUAL
操作権	T/B	×	×	× 注 2)	○
	コントローラ操作パネル	○ 注 3)	×	× 注 2)	×
	パソコン	○ 注 3)	×	× 注 2)	×
	外部信号	○ 注 3)	×	× 注 2)	×

注 1）"AUTOMATIC" モード時に T/B で＜操作パネル＞画面を表示している間は、操作権は T/B に移ります。（T/B の状態表示ランプ [ENABLE] が点滅）
この間、その他の操作機器からの操作権を必要とする操作はできませんのでご注意願います。
注 2）T/B が " 有効 " 状態で、コントローラのモードを "AUTOMATIC" にするとエラー 5000 が発生します。
注 3）操作権入力信号：IOENA の設定により有効な機器が変わります。

表 2-2：操作権を必要とする操作　　操作権：○ = 操作権必要、× = 操作権不要

区分	操作権	操作
運転 注 1)	○	サーボオン
	×	サーボオフ
	○	プログラム始動。コントローラ操作パネルの他、T/B の操作でも起動ができます。
	×	プログラム停止 / サイクル停止
	○	プログラムリセット
	×	エラーリセット
	○	オーバーライド変更。但し、T/B からは常時可能。
	×	オーバーライド読み出し
	○	プログラム番号変更
	×	プログラム番号 / 行番号読み出し
入出力信号 注 2)	×	入出力信号読み出し
	×	出力信号書き込み
	○	専用入力 - 始動 / リセット / サーボオン / ブレーキ入切 / 手動モード切替 / 汎用出力リセット / プログラム番号指定 / 行番号指定 / オーバーライド指定
	×	専用入力 - 停止 / サーボオフ / 連続･サイクル / 操作権入力信号 / プログラム番号出力要求 / 行番号出力要求 / オーバーライド出力要求 / エラー番号要求 / 数値入力
	×	ハンド入出力信号読み出し
	○	ハンド出力信号書きこみ
プログラム編集 注 3)	○	行登録 / 読み出し / 呼出し、位置追加 / 補正 / 読み出し、変数書きこみ / 読み出し
	○	ステップ送り / 戻し、実行
	○	ステップアップ / ダウン
	○	ステップジャンプ、ダイレクト実行、ジョグ
ファイル操作	×	プログラム一覧読み出し / 保護設定 / コピー / 削除 / 名称変更 / 初期化
メンテナンス操作	×	パラメータ読み出し、時計設定 / 読み出し、稼働時間計読み出し、アラーム履歴読み出し
	○	原点設定、パラメータ変更

注 1）T/B の＜操作パネル＞から操作する場合、操作権はコントローラのモードにより以下になります。
- "MANUAL".....T/B の [TB ENABLE] を押して有効にする必要があります。
- "AUTOMATIC".....T/B の [TB ENABLE] を押さなくても、＜操作パネル＞画面表示中は、操作権が自動的に T/B に移ります。（T/B の状態表示ランプ [ENABLE] が点滅）

注 2）T/B で＜操作パネル＞画面を表示している間は、[MONITOR] キーによる操作は無効となります。
注 3）ひとつの機器がオンラインで編集中の時、他の機器からの編集はできません。

2.4 動作及び制御関連の機能

本コントローラには表 2-3 に示すような特徴的な機能があります。

表 2-3：特徴的な機能

機能	解説	関連ページ
最適速度制御	ロボットが 2 点間を移動する際にロボットの姿勢によっては速度を抑えなければならないような姿勢がある場合に軌跡の途中での速度を抑え、速度過大エラーを防止する機能です。ただしこの機能が有効時はロボットの先端速度は一定になりません。	269 ページの「Spd（スピード）」を参照してください。
最適加減速制御	ロボットが動き出すときや止まる時の加減速時間に関してハンドの重量や重心、ワークの有無を設定することにより、最適な加減速時間を自動的に設定して動作する機能です。タクトアップにつながります。条件によってはタクトが落ちる場合もあります。	251 ページの「Oadl（オーエーディーエル）」、232 ページの「LoadSet（ロードセット）」を参照してください。
直交コンプライアンス	サーボからのフィードバックデータを元にロボットを柔らかく制御する機能です。ワークの嵌め合いに有効な機能です。ロボットの直交座標系に沿ったならいが可能です。 ただし、ワークの条件によっては使えない場合もあります。	181 ページの「Cmp Tool（コンプツール）」などを参照してください。
衝突検知	ロボットのツールやアームが周辺機器に干渉した場合、緊急停止してツールやアームの破損を最小限に抑えます。自動運転時、ジョグ操作時にも機能を有効にできます。（一部の特殊仕様を除く） 注）本機能はマルチメカ制御と同時に使用することができませんのでご注意ください。	187 ページの「ColChk（コルチェック）」、408 ページの「動作パラメータ」の COL を参照してください。
干渉回避機能	本機能は CR7xx-Q シリーズコントローラにおいて、ロボット CPU 間直接通信を利用して 2 台または 3 台のロボット間の干渉チェックをおこないながらロボットを動作させる機能です。 ジョグ操作や自動運転時にロボット同士の干渉を事前検知して停止させることにより、ロボットへのダメージを軽減させることができます。	481 ページの「5.24 干渉回避機能について」を参照してください。
メンテ予報	ロボットのバッテリー、ベルト、グリスのメンテナンス情報をロボットの動作状態を元に予測します。この機能はオプションのパソコンサポートソフトウェアにてメンテナンス情報が確認できます。（一部の特殊仕様を除く）	オプションのパソコンサポートソフトウエアをご利用ください。
位置復旧支援	関節軸のズレ、モータ交換、ハンド変形、ロボットのベースズレが発生した場合、最大で 10 数ポイントのみ補正することにより原点データやツール、ベースの補正値を計算し位置ズレを補正する機能です。この機能はオプションのパソコンサポートソフトウェアにておこないます。垂直多関節ロボットで使用可能です。（一部の特殊仕様を除く）	オプションのパソコンサポートソフトウエアをご利用ください。
連続軌跡制御	複数の位置を連続的に加減速を抑えてロボットを動作させる機能です。タクトアップに有効な手段です。	104 ページの「(4) 連続動作」、185 ページの「Cnt（コント）」を参照してください。
マルチタスク プログラム運転	ロボットの動作用プログラム、外部機器との通信用プログラムなどに分けて並列に実行できます。入出力処理の短縮に有効です。また、周辺治具の制御用プログラムを作成することによりシーケンサレスのシステム構築も可能です。	127 ページの「4.3.1 マルチタスクとは」、284 ページの「XRun（エックスラン）」などの X*** 命令を参照してください。
プログラムの 常時実行機能	コントローラの電源を ON した時点からプログラムを常時実行させられます。この機能はマルチタスク機能を利用してシーケンサの代わりをロボットプログラムにて行う場合に有効です。	427 ページの「5.4 プログラムパラメータ」の「SLTn」パラメータの起動属性（ALWAYS）を参照してください。
コンティニュー機能	電源 OFF 時の状態を保存し、次回の電源 ON にて電源 OFF 時の状態の続きから再開できます。	427 ページの「5.4 プログラムパラメータ」の「CTN」パラメータを参照してください。
付加軸制御	最大 2 軸までロボットの付加軸として制御可能です。 ロボットのティーチングデータの中に付加軸の位置も記憶しますので完全な同期制御ができます。また付加軸（走行軸）も移動しながらの円弧補間も可能です。	別冊の「付加軸インタフェース取扱説明書」を参照してください。
マルチメカ制御	標準のロボット以外にサーボモータによって組み立てたロボット（ユーザメカ）を最大 2 台（標準ロボット除く）まで制御できます。	別冊の「付加軸インタフェース取扱説明書」を参照してください。
外部機器通信機能	外部機器との通信には、以下の方法があります。 コントローラの制御用及びプログラム内のインターロック機能 1）入出力信号での方法 （CR7xx-Q：シーケンサリンク入出力：8192/8192 点） （CR7xx-D：パラレル入出力：256/256 点（最大）） 2）外部機器とのデータリンク機能（※CR7xx-D のみ） 3)Ethernet での通信 参考：データリンクとは外部機器（ビジョンセンサなど）と補正量などのデータを送受信する機能です）	315 ページの「M_In/M_Inb/M_In8/M_Inw/M_In16」、325 ページの「M_Out/M_Outb/M_Out8/M_Outw/M_Out16」変数を参照してください。

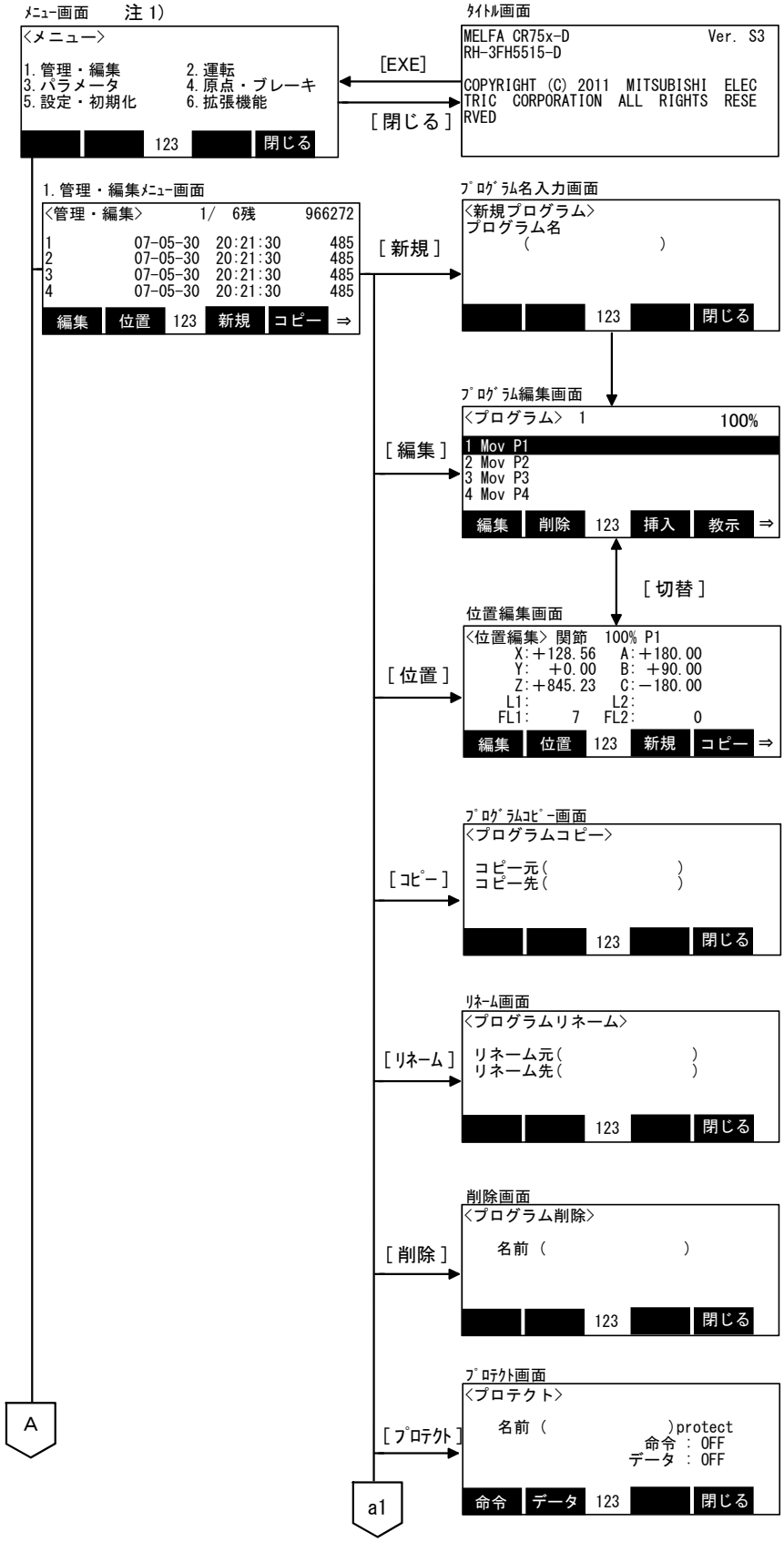
機能	解説	関連ページ
割り込み監視	プログラム動作中に信号などを監視し条件が一致した場合、現在の処理を中断して割り込み処理を実行できます。 ワーク搬送時のワーク落下監視などに有効です。	191 ページの「Def Act（デフ アクト）」、166 ページの「Act（アクト）」を参照してください。
プログラム間ジャンプ	CallP 命令にてプログラムからプログラムを呼び出せます。	171 ページの「CallP（コールピー）」を参照してください。
パレット演算	格子状に配置してあるワークの位置やカセット内のガラス基板の位置を演算によって求めることができます。ティーチング量の軽減に役立ちます。列 × 行形式、1 列形式及び円弧形式が可能です。	110 ページの「4.1.2 パレット演算」、199 ページの「Def Plt（デフ パレット）」、257 ページの「Plt（パレット）」を参照してください。
ユーザ定義領域	最大 32 エリアまで任意の空間を指定し、その領域内にロボットの先端位置が入っているかをリアルタイムに監視し、その状態を外部に出力したり、プログラムでチェックしたり、エラーにしたりできます。 また、ロボットプログラム内で同様な機能を持った関数（Zone、Zone2、Zone3）もあります。	435 ページの「5.8 ユーザ定義領域について」、340 ページの「M_Uar」,341 ページの「M_Uar32」変数を参照してください。 405 ページの「Zone」、406 ページの「Zone2」、406 ページの「Zone3」を参照してください。
関節動作範囲 直交動作範囲 自由平面リミット	ロボットの動作範囲を 3 種類の方法で制限できます。 関節動作範囲：各軸の動作範囲ごとに制限できます。 直交動作範囲：ロボットの直交座標系で動作範囲を制限できます。 自由平面リミット：任意の平面を定義して平面の手前か奥かによって動作範囲を制限できます。	408 ページの「5.1 動作パラメータ」の「MEJAR」、「MEPAR」パラメータを参照してください。 440 ページの「5.9 自由平面リミット」を参照してください。

3 操作方法の解説

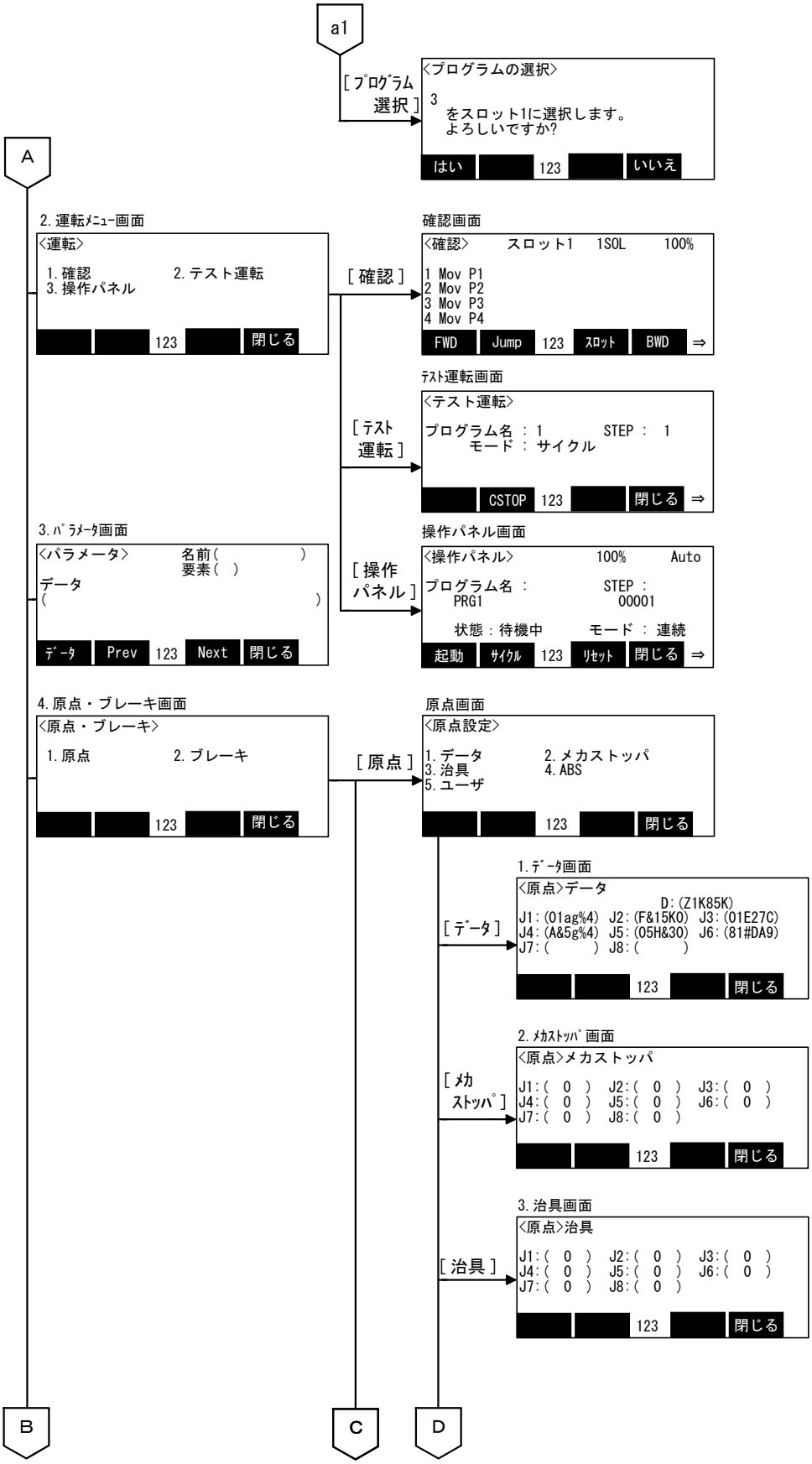
R32TB(オプション) の操作方法を記述します。

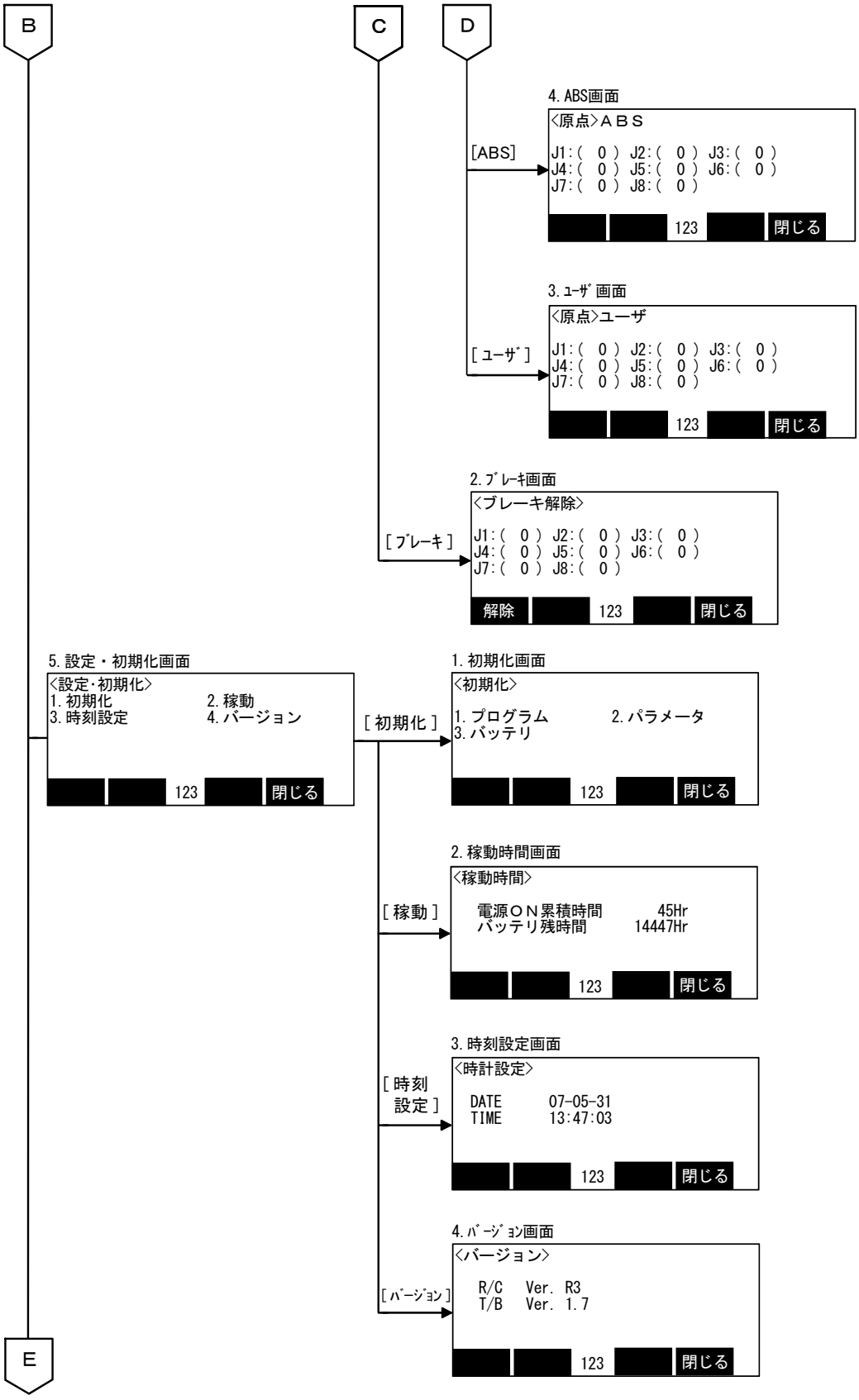
3.1 T/Bのメニュー画面操作

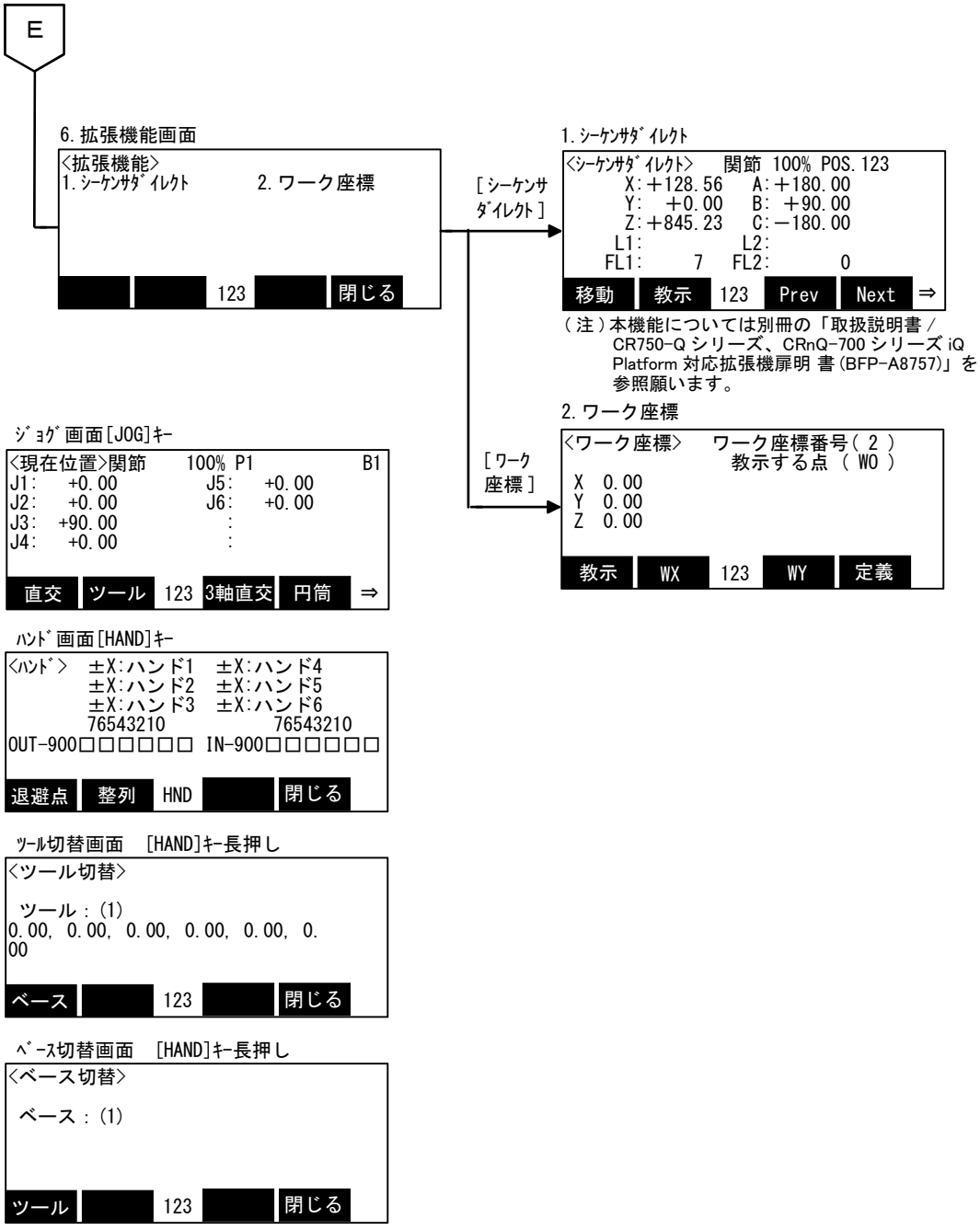
(1) 画面ツリー



注 1)6. 拡張機能中の
"1. ｼｰｹﾝｻﾀﾞｲﾚｸﾄ" については別冊の「取扱説明書 /CR750-Q/CR751-Q シリーズ、CRnQ-700 シリーズ iQ Platform 対応拡張機能説明書 (BFP-A8757)」を参照願います。


a1
[ﾌﾟﾛｸﾞﾗﾑ 選択]
〈プログラムの選択〉
3
をスロット1に選択します。
よろしいですか?
はい
123
いいえ
A
2.運転ﾒﾆｭｰ画面
〈運転〉
1.確認
2.テスト運転
3.操作パネル
123
閉じる
[確認]
確認画面
〈確認〉 スロット1 1SOL 100%
1 Mov P1
2 Mov P2
3 Mov P3
4 Mov P4
FWD
Jump
123
ｽﾛｯﾄ
BWD
⇒
ﾃｽﾄ運転画面
〈テスト運転〉
プログラム名 : 1 STEP : 1
モード : サイクル
CSTOP
123
閉じる
[テスト 運転]
3.ﾊﾟﾗﾒｰﾀ画面
〈パラメータ〉 名前( )
要素( )
データ
( )
ﾃﾞｰﾀ
Prev
123
Next
閉じる
[操作 パネル]
操作パネル画面
〈操作パネル〉 100% Auto
プログラム名 : PRG1
STEP : 00001
状態 : 待機中 モード : 連続
起動
ｻｲｸﾙ
123
ﾘｾｯﾄ
閉じる
⇒
4.原点・ブレーキ画面
〈原点・ブレーキ〉
1.原点
2.ブレーキ
123
閉じる
[原点]
原点画面
〈原点設定〉
1.データ
2.メカストッパ
3.治具
4.ABS
5.ユーザ
123
閉じる
1.ﾃﾞｰﾀ画面
〈原点〉データ
D:(Z1K85K)
J1:(01ag%4) J2:(F&15K0) J3:(01E27C)
J4:(A&5g%4) J5:(05H&30) J6:(81#DA9)
J7:( ) J8:( )
123
閉じる
[ﾃﾞｰﾀ]
2.ﾒｶｽﾄｯﾊﾟ画面
〈原点〉メカストッパ
J1:(0) J2:(0) J3:(0)
J4:(0) J5:(0) J6:(0)
J7:(0) J8:(0)
123
閉じる
[メカ ｽﾄｯﾊﾟ]
3.治具画面
〈原点〉治具
J1:(0) J2:(0) J3:(0)
J4:(0) J5:(0) J6:(0)
J7:(0) J8:(0)
123
閉じる
[治具]
B
C
D


B
C
D
E
4. ABS画面
<原点>ＡＢＳ
[ABS]
J1:(0) J2:(0) J3:(0)
J4:(0) J5:(0) J6:(0)
J7:(0) J8:(0)
123
閉じる
3. ﾕｰｻﾞ 画面
<原点>ユーザ
[ﾕｰｻﾞ]
J1:(0) J2:(0) J3:(0)
J4:(0) J5:(0) J6:(0)
J7:(0) J8:(0)
123
閉じる
2. ﾌﾞﾚｰｷ画面
<ブレーキ解除>
[ﾌﾞﾚｰｷ]
J1:(0) J2:(0) J3:(0)
J4:(0) J5:(0) J6:(0)
J7:(0) J8:(0)
解除
123
閉じる
5. 設定・初期化画面
<設定･初期化>
1. 初期化
2. 稼動
3. 時刻設定
4. バージョン
123
閉じる
[初期化]
1. 初期化画面
<初期化>
1. プログラム
2. パラメータ
3. バッテリ
123
閉じる
[稼動]
2. 稼動時間画面
<稼動時間>
電源ＯＮ累積時間 45Hr
バッテリ残時間 14447Hr
123
閉じる
[時刻
設定]
3. 時刻設定画面
<時計設定>
DATE 07-05-31
TIME 13:47:03
123
閉じる
[ﾊﾞｰｼﾞｮﾝ]
4. ﾊﾞｰｼﾞｮﾝ画面
<バージョン>
R/C Ver. R3
T/B Ver. 1.7
123
閉じる

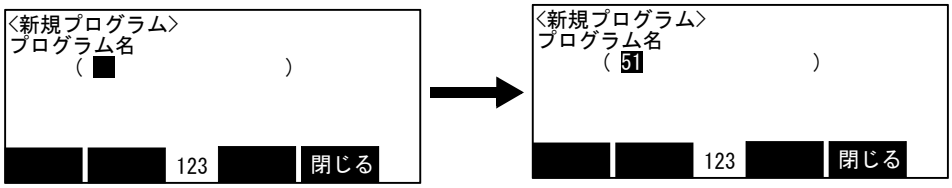

E
6. 拡張機能画面
<拡張機能>
1. ｼｰｹﾝｻﾀﾞｲﾚｸﾄ
2. ワーク座標
123
閉じる
[ｼｰｹﾝｻﾀﾞｲﾚｸﾄ]
1. ｼｰｹﾝｻﾀﾞｲﾚｸﾄ
<ｼｰｹﾝｻﾀﾞｲﾚｸﾄ> 関節 100% POS.123
X:＋128.56 A:＋180.00
Y: ＋0.00 B: ＋90.00
Z:＋845.23 C:－180.00
L1: L2:
FL1: 7 FL2: 0
移動
教示
123
Prev
Next
⇒
（注）本機能については別冊の「取扱説明書 / CR750-Q シリーズ、CRnQ-700 シリーズ iQ Platform 対応拡張機扉明 書(BFP-A8757)」を参照願います。
[ワーク座標]
2. ワーク座標
<ワーク座標> ワーク座標番号(2)
教示する点 (W0)
X 0.00
Y 0.00
Z 0.00
教示
WX
123
WY
定義
ｼﾞｮｸﾞ画面[JOG]ｷｰ
<現在位置>関節 100% P1 B1
J1: +0.00 J5: +0.00
J2: +0.00 J6: +0.00
J3: +90.00 :
J4: +0.00 :
直交
ツール
123
3軸直交
円筒
⇒
ﾊﾝﾄﾞ画面[HAND]ｷｰ
<ﾊﾝﾄﾞ> ±X:ハンド1 ±X:ハンド4
±X:ハンド2 ±X:ハンド5
±X:ハンド3 ±X:ハンド6
76543210 76543210
OUT-900□□□□□□□ IN-900□□□□□□□
退避点
整列
HND
閉じる
ﾂｰﾙ切替画面 [HAND]ｷｰ長押し
<ツール切替>
ツール : (1)
0.00, 0.00, 0.00, 0.00, 0.00, 0.00
ベース
123
閉じる
ﾍﾞｰｽ切替画面 [HAND]ｷｰ長押し
<ベース切替>
ベース : (1)
ツール
123
閉じる

(2) 数字 / 文字の入力

[CHARACTER] キーを押す毎に、数字入力モードと、文字入力モードが切り替わります。
画面右下に現在の入力モードが表示されており、″123″ の表示は数字入力モード、″ABC″ の表示は文字入力モードであることを示しています。

1) 数字の入力

数字 (″-″(マイナス)、″.″(小数点) を含む) は各キーの左下に表示されているキーを押すと入力できます。
[CHARACTER] キーを押して、画面右下に ″123″ を表示させた状態で数字キーを押して入力します。
例) プログラム名に ″51″ を入力する場合。

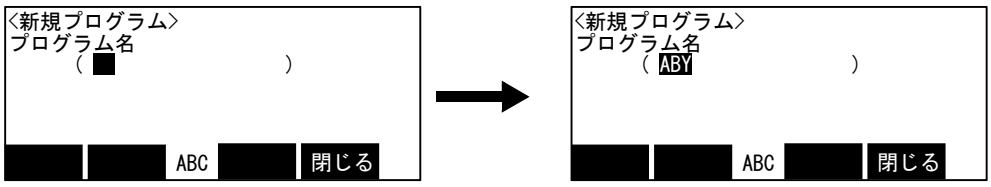


数字の入力 [CHARACTER] [5] [1]

2) 文字の入力

各キーの右下に文字が表示されています。そのキーを押すと文字が入力できます。
[CHARACTER] キーを押して、画面右下に ″ABC″ を表示させた状態で文字キーを押して入力します。
複数の文字が表示されているキーは、そのキーを押す毎に入力文字が変わります。
例)[ABC] キーの場合、そのキーを押す毎に ″A″ ″B″ ″C″ ″a″ ″b″・・・・・・と繰り返します。

同じキーに表示されている文字を続けて入力する場合は、一旦 [→] キーを押してカーソルを進めてください。
例)″ABY″ を入力する場合は、[ABC]、[→]、[ABC] を 2 回、[WXYZ] を 3 回押します。



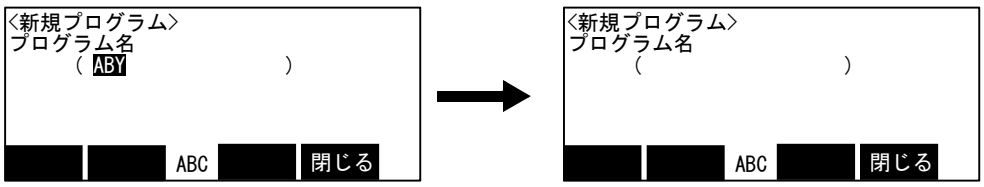
文字の入力 [CHARACTER] [ABC] [→] [ABC] [ABC] [WXYZ] [WXYZ] [WXYZ]

キーに表示されていない文字を入力することができます。表示されていない文字が割り付けられているキーと、そのキーで入力できる文字を以下に示します。

a) [’ ( )] キー..... ’ → (→) → ″ → ^ → : → ; → ¥ →?
b) [@ =] キー..... @ → = → + → - → * → / → 〈 → 〉
c) [, %] キー..... , → % → # → $ → ! → & → _ → .

3) 文字の削除

間違えて入力した文字は、[CLEAR] キーを押すとカーソルの位置にある文字を 1 文字削除します。
例)″ABY″ の ″B″ を ″M″ に変更し、″AMY″ にする場合。
文字 ″B″ にカーソルを合わせて、[CLEAR] キーを押して削除した後、″M″、″Y″ を入力します。



入力文字の修正 [←] [CLEAR] [MNO] [WXYZ] [WXYZ] [WXYZ]

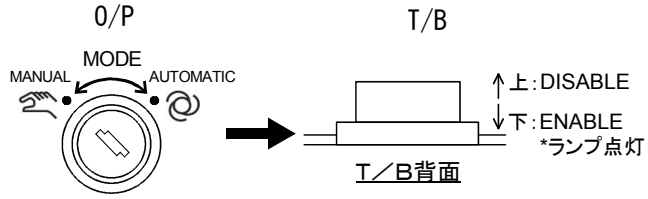
[CLEAR] キーを長押しすると、カッコ内の文字をすべて消去します。

(3) メニューの選択方法

メニューの選択方法には以下の 2 種類があります。いずれかの方法でおこなってください。

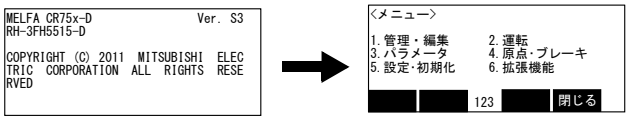
・選択したい項目の数字ｷｰを押す。
・選択したい項目へカーソルを移動させ [EXE] ｷｰを押す。

それぞれの方法でメニュー画面より ″1. 管理・編集 ″ を選択する場合を示します。



1) コントローラのモードを ″MANUAL″ にして、T/B の [ENABLE] スイッチを押して T/B を有効にします。(スイッチと ENABLE LED 点灯)

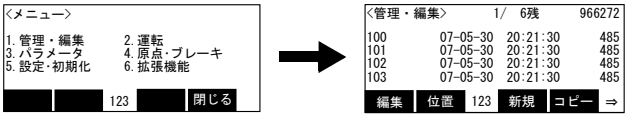
タイトル画面からメニュー画面がでます。



メニュー画面の表示 [EXE]

2) < タイトル > 画面が出た状態でいずれかのｷｰ (例　[EXE] ｷｰ) を押すことにより < メニュー > 画面を表示します。

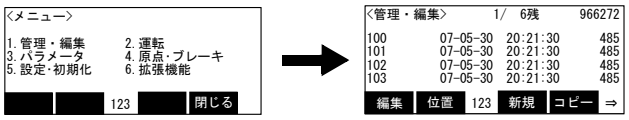
a) 数字キーでメニューを選択する



メニュー画面の表示 [1]

[CHARACTER] ｷｰを押して数字入力モードにし、[1] ｷｰを押します。< 管理・編集 > 画面を表示します。

b) 矢印キーでメニューを選択する



メニュー画面の表示 [↑][↓][←][→]+[EXE]

矢印ｷｰ ([↑]、[↓]、[←]、[→]) を押してカーソルを ″1. 管理・編集 ″ に合わせて [EXE] ｷｰを押します。
< 管理・編集 > 画面を表示します。

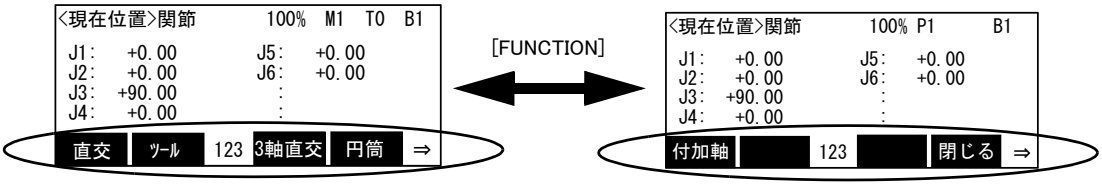
他のメニュー画面でも同様の操作でおこないます。

◇◆◇ T/B を使う時は◇◆◇

コントローラのモードを ″MANUAL″ にします。それ以外では特定の操作 (ジョグ画面で現在位置表示、オーバーライド変更、モニタ) のみ可能です。

◇◆◇ファンクションキー◇◆◇

画面の最下段に白抜き文字で表示されたメニューがあります。これらは左から順に [F1]、[F2]、[F3]、[F4] ｷｰに割り当てられています。対応するファンクションキーを押すことで表示されているメニューを選択することができます。
また、メニューの右端に ″⇒″ が表示されている場合は、現在の表示の他にもまだメニューがあることを示しており、[FUNCTION] ｷｰを押すごとに表示メニューが切り替わります。



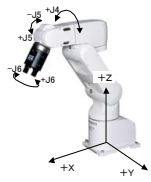
3.2 ジョグ送り(概略説明)

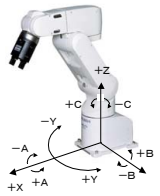
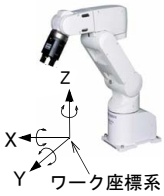
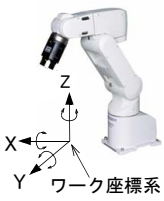
ジョグ送りとはロボットを手動で動かす操作のことをいいます。
ここでは垂直多関節型ロボットを元に概略を説明をします。
ロボットの機種によって軸の構成が異なりますので、個々の機種に関しては別冊の「ロボット本体のセットアップから保守まで」編をご覧ください。

3.2.1 ジョグ送りの種類

ジョグ送りには以下の 6 種類があります。(以下、ロボットの絵は垂直多関節型 6 軸タイプを例に示しています。)

表 3-1 : ジョグ送りの種類

種類	操作	説明
関節ジョグ 	1) [TB ENABLE] スイッチを ENABLE にします。 2) イネーブルスイッチ (3 ポジションスイッチ) を押さえます。 3) [SERVO] キーを押してサーボ ON します。 4) [JOG] キーを押して [F1](関節) キーを押して関節ジョグモードを選択します。 5) 各軸のキーを押します。 J1 ～ J6 軸。	ロボットを各軸ごとに移動させます。 J1 ～ J6 軸までと付加軸 J7, J8 がそれぞれ独立して動かせます。 軸数はロボットの機種によって異なります。 付加軸キーは [J1] キーが J7 軸、[J2] キーが J8 軸に対応します。
直交ジョグ 	1) 1) ～ 3) は関節ジョグと同じです。 4) " 直交 " に対応するいずれかのファンクションキー ([F1] から [F4]) を押して直交ジョグモードを選択します。 5) 各軸のキーを押します。 X,Y,Z,A,B,C 軸。	X, Y, Z のキーを押すと、ロボットのワールド座標系を基準に X, Y, Z 方向にハンドの向きを維持したまま直線で動作します。 A, B, C のキーを押すとワールド座標系の X, Y, Z 軸を中心に回転し、ハンドの向きを変えます。先端の位置は固定です。 ただし、パラメータ MEXTL にてツール長を正しく設定しておく必要があります。
ツールジョグ 	1) 1) ～ 3) は関節ジョグと同じです。 4) " ツール " に対応するいずれかのファンクションキー ([F1] から [F4]) を押してツールジョグモードを選択します。 5) 各軸のキーを押します。 X,Y,Z,A,B,C 軸。	X, Y, Z のキーを押すと、ロボットのハンド先端を原点とするツール座標系を基準にハンドの向きを維持したまま、前後、左右、上下方向に直線で移動します。 A, B, C のキーを押すとツール座標系の X, Y, Z 軸を中心に回転し、ハンドの向きを変えます。先端の位置は固定です。ただし、パラメータ MEXTL にてツール長を正しく設定しておく必要があります。 ハンド先端のツール座標系はロボットの機種によってことなります。 垂直多関節型のロボットはメカニカルインタフェース面からハンド先端方向が +Z になります。 水平多関節型のロボットはメカニカルインタフェース面から上方向が +Z になります。
三軸直交ジョグ 	1) 1) ～ 3) は関節ジョグと同じです。 4) "3 軸直交 " に対応するいずれかのファンクションキー ([F1] から [F4]) を押して 3 軸直交ジョグモードを選択します。 5) 各軸のキーを押します。 X,Y,Z,A,B,C 軸。	X, Y, Z のキーを押すと、ロボットのワールド座標系を基準に X, Y, Z 方向に直線で動作します。 A, B, C のキーを押すと関節ジョグと同様に関節を基準に動作させますが、制御点の位置 (XYZ 値) を保持するために、姿勢が変わります。(直交ジョグとは異なり、X, Y, Z, J4, J5, J6 にて補間しますので姿勢は維持しません) パラメータ MEXTL にてツール長を正しく設定しておく必要があります。

種類	操作	説明
円筒ジョグ  	1) 1) ～ 3) は関節ジョグと同じです。 4) ″円筒″に対応するいずれかのファンクションキー([F1]から[F4])を押して円筒ジョグモードを選択します。 5) 各軸のキーを押します。 X, Y, Z, A, B, C 軸。	ロボットのワールド座標系の原点を基準に円筒方向にハンドを動作させる場合に使用します。 X 軸はワールド座標系の原点から放射線方向に移動します。 Y 軸は J1 軸を回転させる関節ジョグと同様な動きになります。(円弧動作) Z 軸は直交ジョグの Z と同じです。(上下動作) A, B, C のキーを押すと直交ジョグと同じ動作をします。水平多関節型ロボットで有効な場合があります
ワークジョグ (ワークジョグモード)  	1) 1) ～ 3) は関節ジョグと同じです。 4) ″ワーク″に対応するいずれかのファンクションキー([F1]から[F4])を押してワークジョグモードを選択します。 5) 各軸のキーを押します。 X,Y,Z,A,B,C 軸。	このジョグ操作を行う場合は、パラメータ WKnJOGMD (n=1 ～ 8) を「0(ワークジョグモード)」に設定しておく必要があります。 X, Y, Z のキーを押すと、予め設定されたロボットのワーク座標系を基準にその X, Y, Z 軸方向にハンドの向きを維持したまま直線で動作します。 A, B, C のキーを押すと同ワーク座標系の X, Y, Z 軸を中心に回転しハンドの向きを変えます。先端の位置は固定です。 ただし、パラメータ MEXTL にてツール長を正しく設定しておく必要があります。 注) ワーク座標系：お客様で事前に設定します。 (最大 8 種類設定可能) ワーク座標系が設定されていない場合は直交ジョグでの動作になります。
ワークジョグ (Ex-T ジョグモード) 本機能は以下のソフトウェアバージョンで使用可能です。 F-Q シリーズ： R5 以降 F-D シリーズ： S5 以降  	1) 1) ～ 3) は関節ジョグと同じです。 4) ″ワーク″に対応するいずれかのファンクションキー([F1]から[F4])を押してワークジョグモードを選択します。 5) 各軸のキーを押します。 X,Y,Z,A,B,C 軸。	このジョグ操作を行う場合は、パラメータ WKnJOGMD (n=1 ～ 8) を「1(Ex-T ジョグモード)」に設定しておく必要があります。 X, Y, Z のキーを押すと、予め設定されたロボットのワーク座標系を基準にその X, Y, Z 軸方向にハンドの向きを維持したまま直線で動作します。 A, B, C のキーを押すと同ワーク座標系の X, Y, Z 軸を中心に回転するようにハンドの向きを変えながら制御点を移動させます。 ただし、パラメータ MEXTL にてツール長を正しく設定しておく必要があります。 注) ワーク座標系：お客様で事前に設定します。 (最大 8 種類設定可能) ワーク座標系が設定されていない場合は XYZ は直交ジョグでの動作になります。ABC はロボットの原点座標系を中心に回転するような動作となります。

なお、表 3-1 の種類に示したジョグの内、直交ジョグ、ツールジョグ、円筒ジョグ、ワークジョグ動作時にロボットの制御点が特異点へ近づくと、ブザー音を発すると共にティーチングボックスの画面に警告マークを表示し、ロボットを操作している人に注意を促します。本機能は、パラメータ(MESNGLSW)によって、有効 / 無効の設定が可能です。(408 ページの「5 パラメータでの設定機能」参照)
また、本機能の詳細を 458 ページの「5.17 特異点近傍警告について」に示していますので、参照願います。

3.2.2 ジョグ送りの速度

[OVRD ↑]/[OVRD ↓] キーで速度を変更することができます。現在の設定速度は画面の右上に % で表示されています。

ジョグ送りの速度は下記の種類があります。

← 「OVRD ↓」キー 「OVRD ↑」キー →

LOW	HIGH	3%	5%	10%	30%	50%	70%	100%

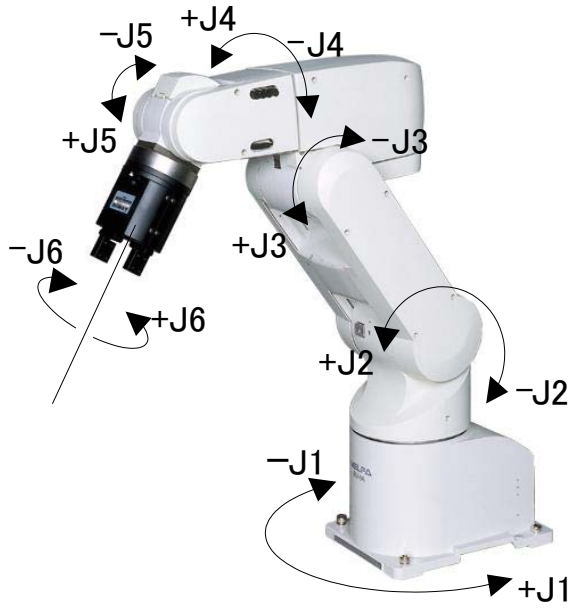
LOW と HIGH は定寸送りになります。定寸送りはキーを 1 回押すと一定量ロボットが動作します。移動量はロボットによって異なります。

表 3-2：定寸量の一例

	関節ジョグ	ツール、直交ジョグ
LOW	0.01 度	0.01mm
HIGH	0.10 度	0.10mm

3.2.3 関節ジョグ

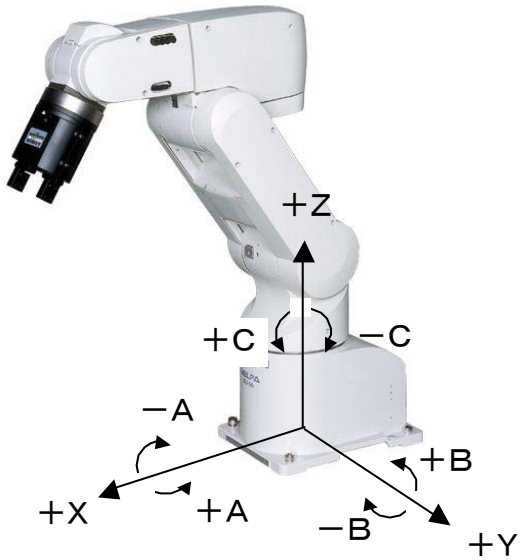
各軸ごとに角度単位で軸を動作させます。



3.2.4 直交ジョグ

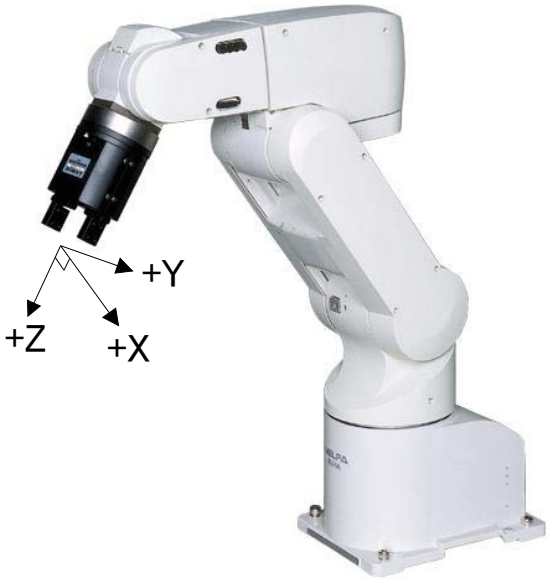
ワールド座標系の向きに沿って移動します。

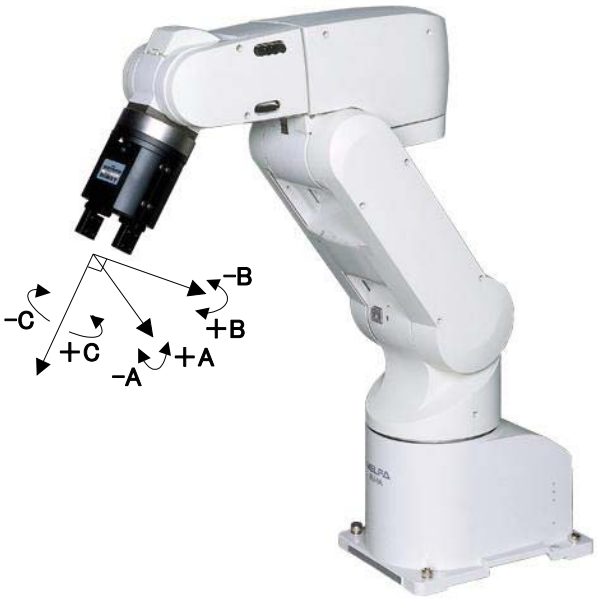
XYZ は mm 単位で移動させます。ABC は角度単位で移動します。



3.2.5 ツールジョグ

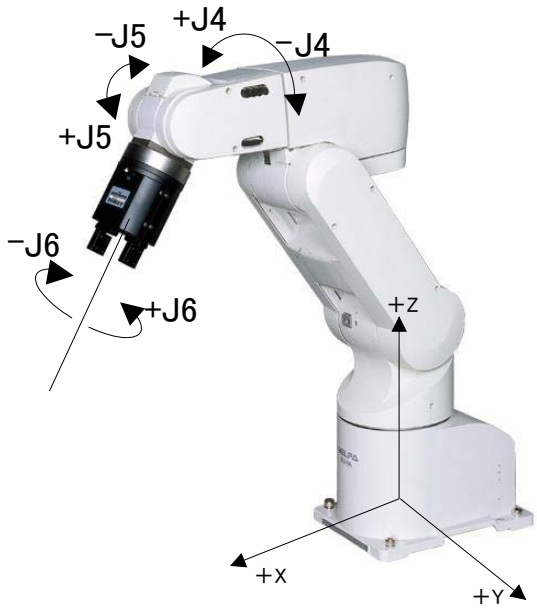
ハンド先端の向きに沿って移動します。
XYZ は mm 単位で移動させます。ABC は角度単位で移動します。





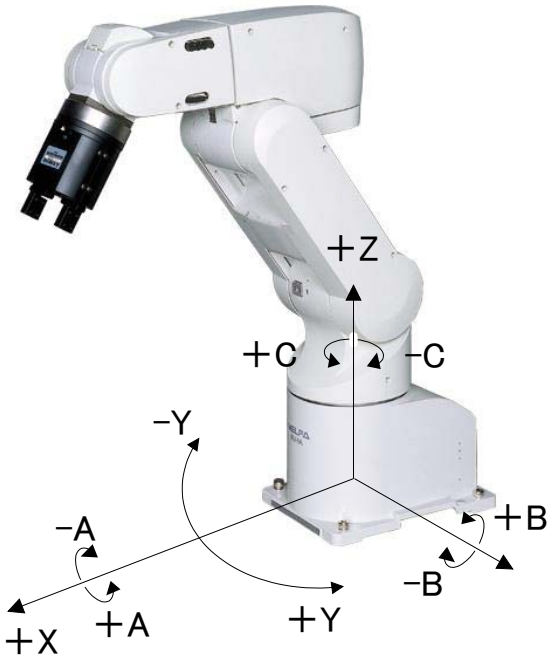
3.2.6 三軸直交ジョグ

XYZ は直交ジョグと同様でワールド座標系の向きに沿って移動します。ABC は関節ジョグと同様に関節を基準に動作させますが、制御点の位置（XYZ 値）を保持するために、姿勢が変わります。
XYZ は mm 単位で移動させます。ABC は角度単位で移動します。



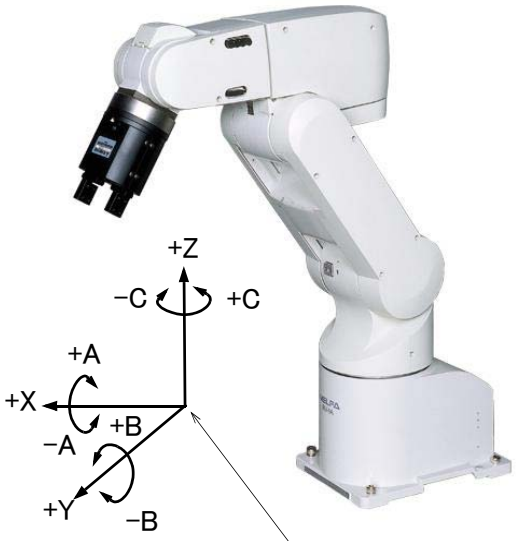
3.2.7 円筒ジョグ

X はロボットのワールド座標系の原点から放射状に遠ざかる方向 / 近づく方向に移動します。Y は J1 軸を中心に回転します。Z はワールド座標系の Z 方向に移動します。ABC 軸は直交ジョグと同じ動きになります。X,Z は mm 単位で移動させます。Y、A、B、C は角度単位移動します。



3.2.8 ワークジョグ

予め設定したワーク座標系 (対象ワークに合わせた座標系) の向きに沿って移動します。
XYZ は mm 単位で移動させます。ABC は角度単位で移動します。

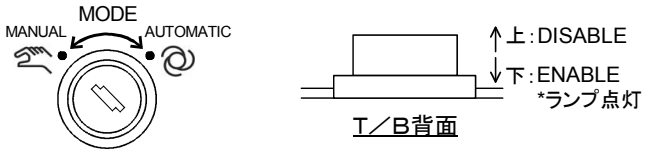


ワーク座標系 : ワークや作業台などに合わせた座標系

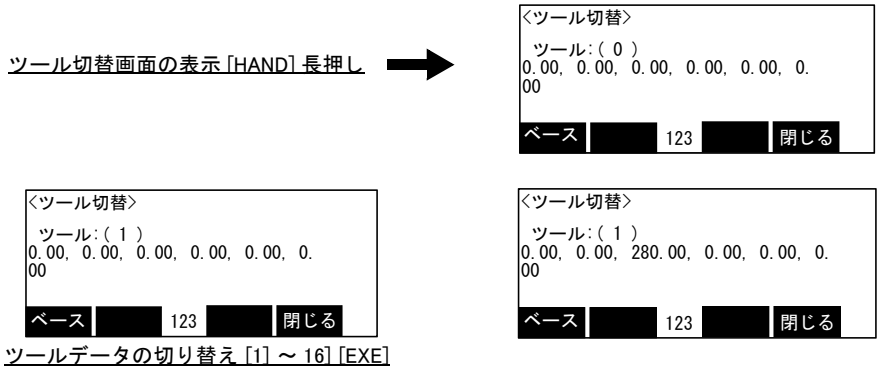
3.2.9 ツールデータの切替

複数のハンドをご使用になる場合は、それぞれのツールデータをあらかじめ登録しておき、ハンドに応じた正しいツールデータを選択してご使用になることが必要です。パラメータでは 16 種類まで登録しておくことができます。
使用するツールのデータをパラメータ：MEXTL1 ～ 16 に設定しておき、以下の操作により使用するツール番号で選択します。

1) コントローラののモードを〝MANUAL〟にして、T/B の [ENABLE] スイッチを押して T/B を有効にします。(スイッチと ENABLE LED 点灯)

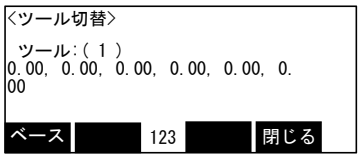


2) [HAND] キーを長押しして、< ツール切替 > 画面を表示させます。
3) 希望する数字キーを押して、[EXE] キーを押すとツールデータが切り替わります。パラメータの MEXTL1 ～ 16 は数字の 1 ～ 16 に対応しています。



ツールデータの切り替え [1] ～ 16] [EXE]

4) 〝閉じる〟に割り当てられたファンクションキーを押して、終了します。



終了 [F4]

5) ジョグ画面の右上に現在のツール番号 (T1 ～ T16) が表示されます。

⚠ 注意 プログラムの自動運転時に、ツールデータ (MEXTL1 ～ 16 パラメータ) を切り替えながらティーチングした位置へ動作させる場合は、必要な箇所で M_TOOL 変数にツール番号を代入してツールデータを切り替えて運転してください。ティーチング時のツールデータと運転時のツール番号が一致しないと思わぬ方向へ動作しますので十分ご注意ください。

⚠ 注意 プログラムのステップ運転時に、ツールデータを切り替えて動作させる場合、ティーチング時のツールデータとステップ運転時のツール番号が一致しないと思わぬ方向へ動作しますので十分ご注意ください。

◇◆◇ツール番号の確認◇◆◇
現在のツール番号は、ツール切替画面、ジョグ画面、または M_Tool 変数で確認できます。

◇◆◇関連情報◇◆◇
MEXTL,MEXTL1,MEXTL2,MEXTL3,MEXTL4.....MEXTL16 パラメータ
Tool 命令、M_Tool 変数
MEXTL パラメータはその時点でのツールデータを保持するパラメータです。MEXTL1 ～ 16 を使用する場合はツール番号を選択した時点で MEXTL パラメータは上書きされますのでご注意ください。
また、ツール番号を 0 に戻すには Tool 命令を実行してください

3.2.10 ワールド座標系の切替(ベース座標番号の指定)

ロボットの位置制御の基準であるワールド座標系を、T/B からの操作で容易に変更することができます。

ベース変換機能 (Base 命令) をご使用の場合、教示作業時に便利な機能です。

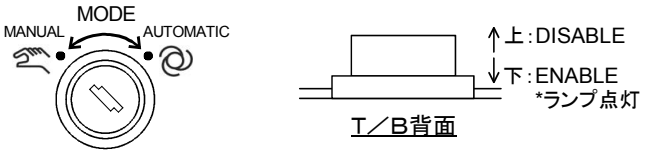
指定するベース座標系を予めパラメータ WK1CORD ～ WK8CORD に設定しておきます。(別冊の「ロボット本体セットアップから保守まで」の 2.4 動作の確認 (6) ﾜｰｸｼﾞｮｸﾞ操作、および「機能と操作の詳細解説」の「5.1 動作パラメータ」参照)

⚠ 注意

本機能でワールド座標系を切り替えると、ロボットは動きませんが現在座標値が変化します。
位置の教示に当たっては、プログラム上で使用している Base 命令に対応した教示位置 (位置変数) と、現在のジョグ操作で設定しているベース座標番号が正しいことを確認してください。
間違ったベース座標番号で教示をおこなうと、プログラム実行時にロボットは予期せぬ動作しをし、物損や人身事故の原因となります。
ベース座標番号と対応する位置変数名を関係付けて間違いなく管理するようお願いします。

以下に操作手順を説明します。

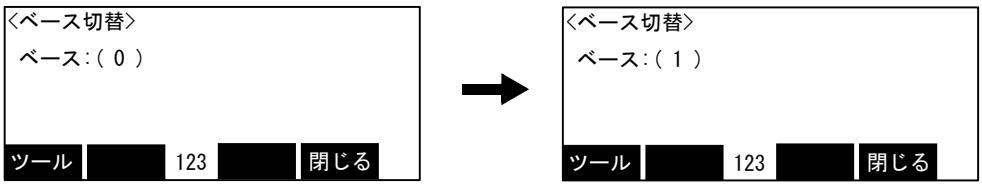
1) コントローラのモードを ″MANUAL″ にして、T/B の [ENABLE] スイッチを押して T/B を有効にします。(スイッチと ENABLE　LED 点灯)



2) [HAND] キーを長押しして < ベース切替 > 画面を表示させます。
< ツール切替 > 画面が表示された場合は、画面下の ″ ベース ″ に該当するファンクションキー [F1] を押します。

< ベース切替 > 画面の表示 : [HAND] キー長押し

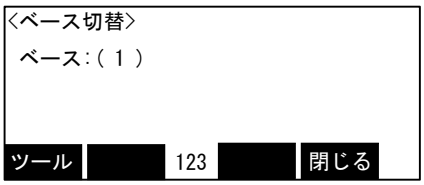
3) 希望するベース座標番号を入力して、[EXE] キーを押すとワールド座標系が切り替わります。
1 ～ 8 : ﾍﾞｰｽ座標番号 (ﾊﾟﾗﾒｰﾀ WK1CORD ～ WK8CORD に対応)
0 : 出荷時の状態に戻す。(ﾍﾞｰｽ変換無しの状態)



(図はﾍﾞｰｽ座標番号 1 に切り替える例)

ﾍﾞｰｽ座標番号によるワールド座標系の設定 [0],[1] ～ [8],[EXE]

4) ″ 閉じる ″ に割り当てられたファンクションキーを押して、終了します。



終了 [F4]

5) ジョグ画面の右上に現在のベース座標番号 (B0 ～ B8) が表示されます。

3.2.11 ジョグ操作中の衝突検知

衝突検知機能を備えている機種があり、パラメータでその有効 / 無効を切り替えることができます。衝突を検知するとエラー番号 101n(下 1 桁の n は軸番号) が発生します。

ジョグ操作時も、本機能を有効にすることができますが、機種によって初期設定状態が異なります。

表 3-3 : 衝突検知パラメータ

パラメータ	名称	要素数	内容	初期値
衝突検知[注 1)]	COL	整数 3	衝突検知の使用可 / 不可と電源 ON 直後の有効 / 無効を定義します。 要素 1 : 衝突検知機能の使用可 (1)/ 不可 (0) 要素 2 : プログラム運転時初期状態として有効 (1)/ 無効 (0) 要素 3 : ジョグ操作中有効にする (1)/ しない (0)/NOERR モード (2) NOERR モードは衝突を検知してもエラーを表示しません。サーボオフするのみとなります。衝突を検知した時にエラーが頻繁に発生して操作しづらい場合にご利用ください。 プログラム運転時 (ポジションジャンプ、ステップ送り含む) 以外は、ジョグ操作用 (要素 3) の設定に依存します。	RH-3FH/6FH/12FH/20FHシリーズ: 1,0,1 RV-Fシリーズ: 0,0,1 RH-3FHRシリーズ: 1,1,1
ジョグ時の検知レベル	COLLVLJG	整数 8	ジョグ操作時 (中断中を含む) の検知レベル (感度) を関節軸毎に設定します。単位 (%) 検知レベル (感度) をあげる場合は数値を小さくしてください。 ジョグ操作中衝突していないのに衝突検知エラーが発生する場合は数値を大きくしてください。 設定範囲 1 ～ 500(%)	機種により異なります。
ハンド条件	HNDDAT0	実数 7	ハンドの初期条件を設定します。(ツール座標系で指定します) 電源ON直後のジョグ操作時はこの設定値が採用されます。 ジョグ操作時に衝突検知を使用される場合は実際のハンド条件を設定してください。設定されないと正しく検知されない場合があります。 (質量, 大きさX, 大きさY, 大きさZ, 重心X, 重心Y, 重心Z) 単位:Kg, mm	機種により異なります。
ワーク条件	WRKDAT0	実数 7	ワークの初期条件を設定します。(ツール座標系で指定します) 電源ON直後のジョグ操作時はこの設定値が採用されます。 (質量, 大きさX, 大きさY, 大きさZ, 重心X, 重心Y, 重心Z) 単位:Kg, mm	0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0

注 1) マルチメカ制御と同時に使用することはできません。

(1) ジョグ操作時の衝突検知レベル調整

ジョグ操作時の衝突検知の感度は低めに設定してあります。より高い衝突検知感度を求められる場合はパラメータの COLLVLJG を調整してお使いください。またパラメータ HNDDAT0 および WRKDAT0 も正しく設定してご使用ください。正しい設定をしないでジョグ操作をおこなうと姿勢によっては誤検知する可能性があります。

◇◆◇衝突検知機能の注意点◇◆◇

衝突検知機能を有効にしていても周辺機器との干渉によるロボット、ハンド、ワーク等の損傷を 100% 阻止できるものではありません。基本的には周辺機器に干渉しないよう気をつけてロボットを操作してください。

◇◆◇衝突後の操作◇◆◇

ハンドやアームが周辺機器と干渉した状態のままサーボオンすると再度衝突検知状態になりサーボオンできない場合があります。何度サーボオンを繰り返してもエラーになる場合は一旦ブレーキ解除操作でアームを逃がしてからサーボオンするか、61 ページの「3.10 解除できないエラーの一時的エラーリセット操作」をおこなって、サーボオンしてからジョグ操作でアームを逃がしてください。

◇◆◇自動運転用の衝突検知との関係◇◆◇

ジョグ操作用の衝突検知と自動運転用の衝突検知の設定は独立しています。

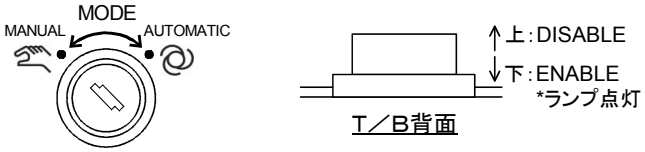
プログラム運転をしていない場合は、ジョグ操作用の設定が採用されます。ジョグ操作用の設定が有効の場合にプログラム中で自動運転用の衝突検知を無効に設定しても、運転を中断するとジョグ操作用の設定 (衝突検知有効) に切り替わります。

3.3 ハンド開閉

ロボットに取り付けたハンドの開閉操作を説明します。

ハンドは最大 6 個制御することができます。　X，Y，Z，A，B，C 軸のキーそれぞれにハンド 6，5，4，3，2，1 が割り当てられ、ハンド開は "+" 側のキー、ハンド閉は "−" 側のキーを押すことで操作できます。

1) コントローラのモードを "MANUAL" にし、T/B の [ENABLE] スイッチを押して T/B を有効にします。（スイッチと ENABLE LED 点灯）

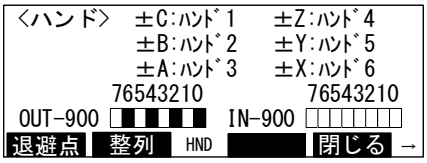


2) 「HAND」キーを押して＜ハンド＞画面を表示させます。

OUT-900 にはハンドの開閉状態を、IN-900 にはハンドチェック入力信号の ON/OFF 状態を示しています。

ハンド 1 を開く場合は [+C] キーを、閉じる場合は [-C] キーを押します。その他のハンドは、X，Y，Z，A，B 軸のキーで同様に操作できます。

ハンド画面の表示 [HAND]



OUT-900 ～ OUT-907	7	6	5	4	3	2	1	0
開 / 閉	閉	開	閉	開	閉	開	閉	開
ハンド番号	4		3		2		1	

IN-900 ～ IN-907	7	6	5	4	3	2	1	0
入力信号番号	907	906	905	904	903	902	901	900

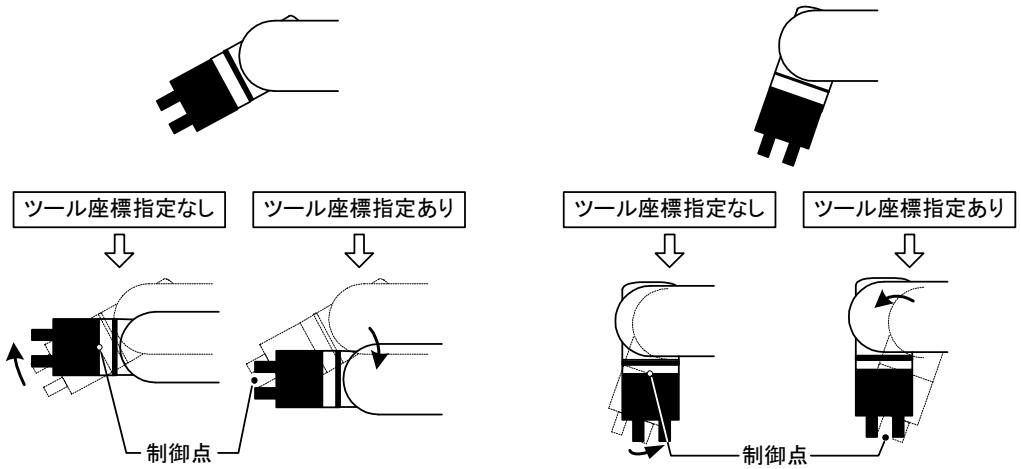
◇◆◇自動運転用の衝突検知との関係◇◆◇

ロボットのハンド部分にはいろいろなツールを取り付けることができます。エアー制御で電磁弁にダブルソレノイドを使う場合は、ハンド開閉操作でハンド信号の 2 ビットを制御します。

ハンド信号の詳細は 444 ページの「5.12 ハンドタイプについて」、および 445 ページの「5.13 ハンド初期状態について」を参照してください。

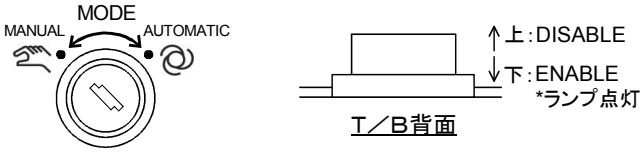
3.4 ハンド整列

ロボットに取り付けたハンドの姿勢を 90 度単位の姿勢に整列することができます。
この機能は、現在位置の A、B、C 成分が 90 度単位の一番近い値にロボットが移動します。



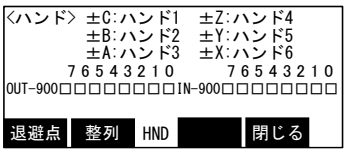
TOOL 命令やパラメータでツール座標が指定されている場合は、指定されたツール座標でハンド整列されます。ツール座標が指定されていない場合は、メカニカルインタフェース中心でハンド整列されます。上図は、垂直多関節型ロボットの例で、[ツール座標指定あり] は、ハンド先端にツール座標を指定した場合です。ツール座標については、432 ページの「5.6 標準ツール座標について」を参照ください。
ハンド整列の手順は以下の通りです。

1) コントローラのモードを ″MANUAL″ にし、T/B の [ENABLE] スイッチを押して T/B を有効にします。（スイッチと ENABLE LED 点灯）

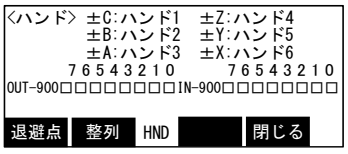


2) イネーブルスイッチ (3 ポジションスイッチ) を押さえて、[SERVO] キーを押し、サーボオンします。
3) 「HAND」キーを押して < ハンド > 画面を表示させます。

ハンド画面の表示 [HAND]



4) イネーブルスイッチ (3 ポジションスイッチ) を押さえたまま、″ 整列 ″ に割り付けられているファンクションキーを押し続けます。押し続けている間、ロボットはハンド整列動作をおこない、動作中はコントローラの [START]LED が点灯します。
動作の途中でいずれかを離すとロボットは停止します。



ハンド整列の実行 [整列]

⚠ 注意 ハンド整列の操作により、姿勢成分 (A、B、C) が 180 度となる場合、同じ姿勢であっても成分の値は +180 度、または -180 度のいずれかとなります。これは内部的な演算誤差によるもので、どちらの値になるかの規則性はありません。この位置をパレット定義命令 (Def Plt) の位置データとして使用する場合、同一の姿勢成分に +180 度と -180 度が混在していると、パレット演算では -180 ～ +180 度の間を分割して位置を算出するため、ハンドが回転し予期せぬ動作となってしまいます。姿勢成分が 180 度の位置データをパレットの定義に使用する場合は、180 度の符号を + または - のいずれかに揃えてください。なお、この位置データを補間命令の目的位置として直接使用する場合は、どちらの値であっても問題なく動作します。

3.5 プログラミング

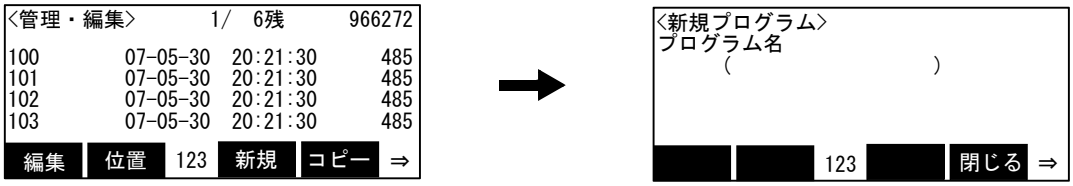
本ロボット CPU で使用している MELFA-BASIC V は、豊富な演算機能により高度な作業を記述することができます。なお、ここでは T/B を使ったプログラミング方法を説明します。MELFA-BASIC V の命令、記述などについては、本書 /164 ページの「4.13 命令の詳細説明」の該当頁参照ください。

3.5.1 プログラムの作成

(1) プログラム編集画面のオープン

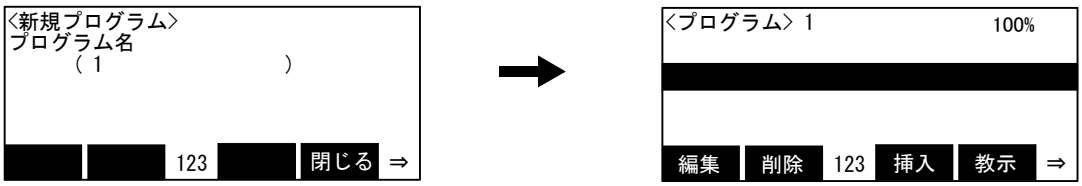
新規に作成するプログラムの編集画面を開きます。以下に、プログラム名 "1" をオープンする例を示します。

1) < メニュー > 画面で "1. 管理・編集 " 画面を選択します。
2) " 新規 " に対応するファンクションキーを押します。プログラム名入力画面を表示します。



メニューの選択 [F3]

3) プログラム名を入力します。命令編集画面を表示します。
(既存のプログラム名を入力した場合は既存のプログラムをオープンします)



プログラム番号 1 の入力 [1] [EXE]

◇◆◇常時実行プログラム (ALWAYS 属性) の編集◇◆◇

SLTn パラメータにて常時実行 (ALWAYS) 属性に設定してあるプログラムを編集するには一度常時実行の属性を解除してから編集してください。常時実行属性のプログラムは常に実行されていますので編集ができません。SLTn パラメータにて ALWAYS を START に変更しコントローラの電源を再投入して常時実行を停止させてください。

◇◆◇既存のプログラムをオープンする◇◆◇

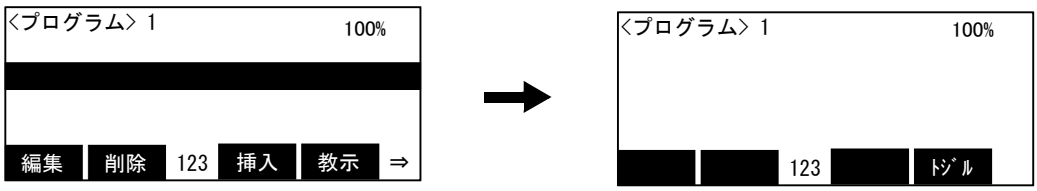
上記のように、プログラム名に既存のプログラム名を入力する方法がありますが、< 管理・編集 > 画面で [↑]、[↓] キーを押して、オープンするプログラムを選んだ状態で " 編集 " に割り当てられているファンクションキーを押すことでも可能です。

(2) プログラムの作成

下記、3 ステップのプログラムを入力する場合のキー操作を示します。

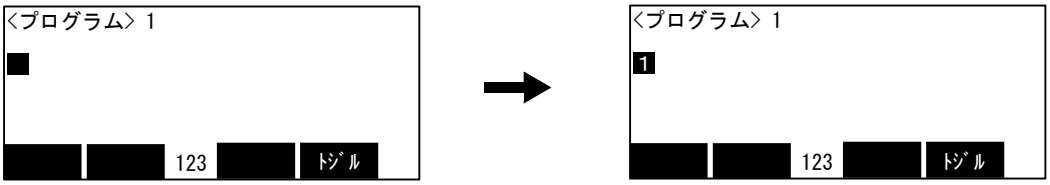
1 Mov P1
2 Mov P2
3 End

1) 命令編集画面で、″ 挿入 ″ に対応するファンクションキー ([F3]) を押します。



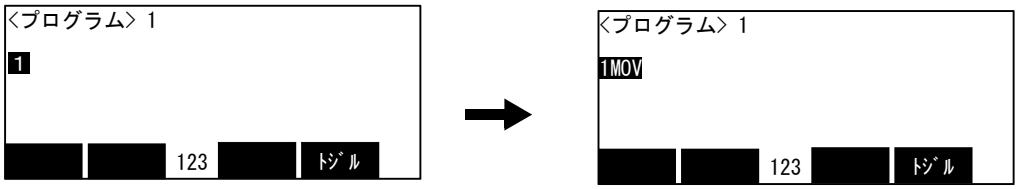
ステップの挿入 [F3]

2) ステップ番号 ″1″ の入力
[CHARACTER] キーを押して数字入力モードにし、[1] キーを押します。
ステップ番号と命令後の間のスペースは省略できます。



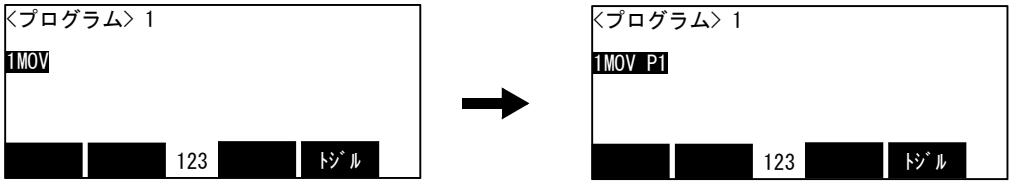
ステップ番号の入力 [1]

3) 命令語 ″Mov″ の入力
[CHARACTER] キーを押して文字入力モードにし、
[MNO] (″M″)、[→]、[MNO] を 3 回 (″O″)、[TUV] を 3 回 (″V″)、[→] キーの順に押します。



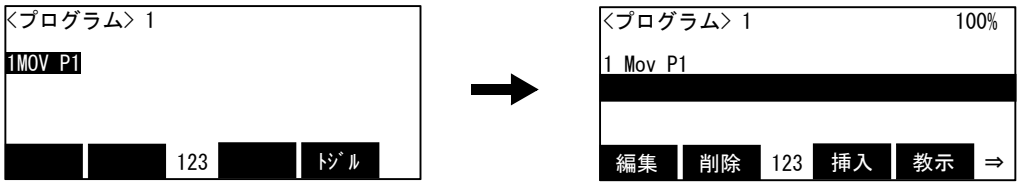
Mov の入力 [CHARACTER] [MNO] [→] [MNO] [MNO] [MNO] [TUV] [TUV] [TUV]

4) 命令語に付随するデータ ″P1″ の入力
[SP] (″ スペース ″)、[PQRS] (″P″) キーを押します。
[CHARACTER] キーを押して数字入力モードにし、[1] キーを押します。
命令語と命令に付随するデータとの間にはスペースが必要です。



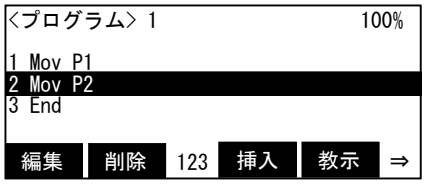
P1 の入力 [SP] [PQRS] [CHARACTER] [1]

5) ステップ 1 の確定
[EXE] キーを押して確定します。



ステップ 1 の確定 [EXE]

6) 以下、同様にステップ 2、3 を入力します。



以上でプログラムの入力ができました。

◇◆◇ステップの前後を表示させるには◇◆◇
[↑]、[↓]キーを押すと、画面をスクロールさせることができます。

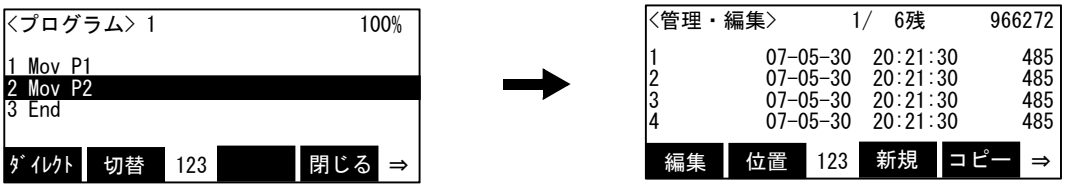
◇◆◇特定ステップを表示させるには◇◆◇
″JUMP″ に対応するファンクションキーを押して表示されるステップ番号入力画面で、希望するステップ番号を入力し、[EXE] キーを押すと、入力されたステップを表示します。

◇◆◇挿入時はステップ番号を省略できます◇◆◇
省略した場合、カーソル行の次へ挿入されます。

◇◆◇大文字と小文字は自動的に変換されます◇◆◇
MELFA BASIC V では、予約語や変数名は大文字と小文字を組み合わせて表示します。T/B からの入力時は、大文字のまま (小文字のまま) でも、行の確認時に自動的に変換します。

(3) プログラム作成の終了、保存

″閉じる″に対応するファンクションキーを押すと、プログラムを保存して作成を終了します。
″閉じる″が表示されていない場合は、[FUNCTION] キーを押すと表示されます。



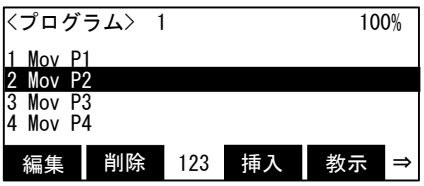
プログラムの終了と保存 [F4]

⚠ 注意 プログラム編集中のままで電源を遮断すると、位置データを含め編集内容は破棄されてしまいますのでご注意願います。
また、できる限りパソコンにも保存し、バックアップをとるようにしてください。
オプションのパソコンサポートソフトウェアにてプログラム管理することをお勧めします。

(4) プログラムの修正方法

プログラムの修正を行う前に、本書 /34 ページの「3.5.1 プログラムの作成」の「(1) プログラム編集画面のオープン」を参照してプログラムの編集画面をオープンしてください。

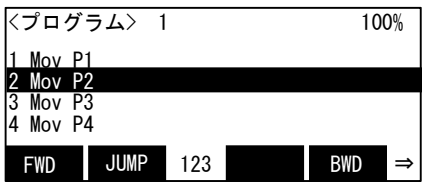
以下に、例としてステップ 5 の ″Mov P5″ を ″Mvs P5″ に修正する場合の操作を示します。



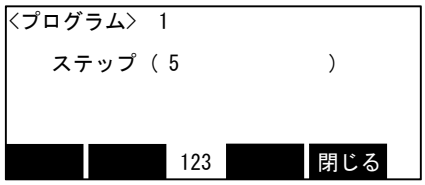
1) ステップ 5 の呼び出し。

″JUMP″ に対応するファンクションキーを押して表示されるステップ番号入力画面で、[5]、[EXE] キーを押します。ステップ 5 にカーソルが移動します。

[↑]、[↓] キーでステップ 5 にカーソルを合わせてもステップ 5 を呼び出すことができます。



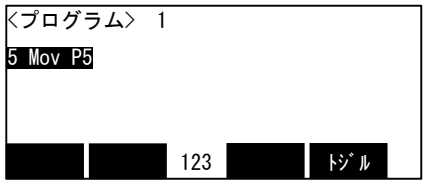
ステップ 5 の呼び出し [F2]



ステップ 5 の呼び出し [5] [EXE]

2) 命令語の修正。

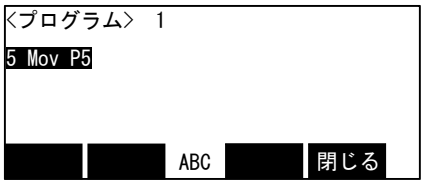
″ 編集 ″ に対応するファンクションキーを押します。

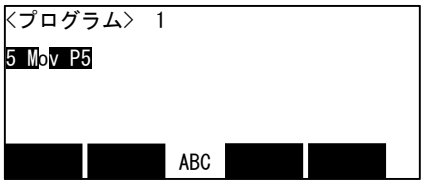


命令語の修正 [F1]

3) [→] キーを 3 回押して、カーソルを ″o″ に合わせます。

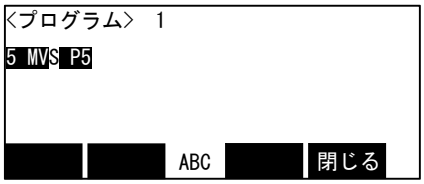
[TUV] キーを 3 回 (″v″ の入力)、[→] キーを 1 回、[PQRS] キーを 4 回 (″s″ の入力) 押すと、″5 Mvs P5″ となりますので、[EXE] キーを押して確定します。

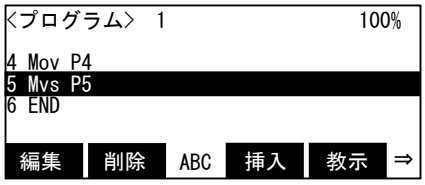




[TUV] [TUV] [TUV] [→]

命令語の修正 [PQRS] [PQRS] [PQRS] [PQRS]





命令語の修正 [EXE]

◇◆◇カーソルを合わせて修正する◇◆◇

[↑]、[↓]キーでステップ 5 にカーソルを合わせて、″ 編集 ″ に対応したファンクションキーを押して修正することもできます。

◇◆◇修正を中止する◇◆◇

修正の途中で、″ 閉じる ″ に対応したファンクションキーを押すと、修正を中止することができます。

◇◆◇文字を修正するには◇◆◇

間違った文字の上へカーソルを移動し、[CLEAR] キーを押して左方向へ削除した後、正しい文字を入力します。

◇◆◇プログラムを修正したら◇◆◇

プログラムを修正したらかならず保存操作 (″ 閉じる ″ に対応するファンクションキー [F4 を押すか、または T/B 背面の [ENABLE] スイッチを押して T/B を無効にする) をし、内容に関してはステップ運転をおこなって、正しく修正されたことを確認してください。

(5) 現在位置の教示

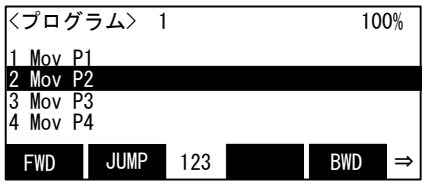
ジョグ操作などでロボットを作業位置へ移動させ、その位置をプログラムで使用している位置変数に教示(登録)することができます。すでに教示済みの場合は上書き(修正)されます。命令編集画面での教示と、位置編集画面での教示があります。

(a) 命令編集画面での教示

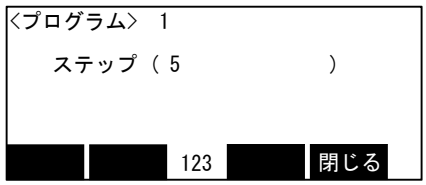
教示する位置変数を使っているステップを呼び出します。
以下に、ステップ 5"Mvs P5" の位置変数 P5 へ現在位置を教示する場合の操作手順を示します。あらかじめジョグ操作などでロボットを作業位置へ移動させておきます。

1) ステップ 5 の呼び出し。

"JUMP" に対応するファンクションキーを押して表示されるステップ番号入力画面で、[5]、[EXE] キーを押します。ステップ 5 にカーソルが移動します。
[↑]、[↓]キーでステップ 5 にカーソルを合わせてもステップ 5 を呼び出すことができます。



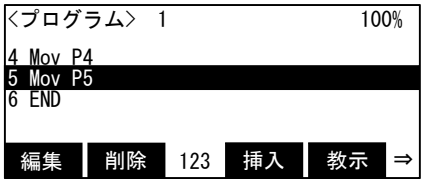
ステップ 5 の呼び出し [F2]



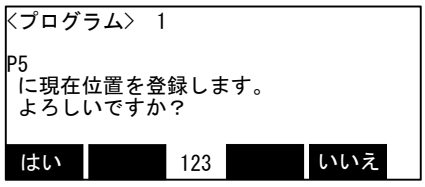
ステップ 5 の呼び出し [5] [EXE]

2) 現在位置の教示

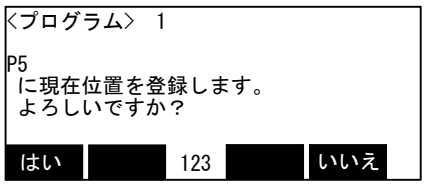
" 教示 " に対応するファンクションキー([F4]) を押すと、確認画面を表示します。



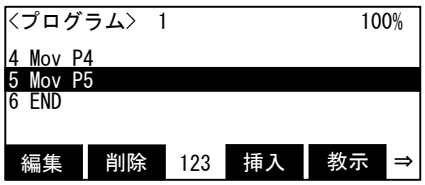
現在位置の登録 [F4]



3) " はい " に対応するファンクションキーを押すと、ロボットの現在位置データが P5 に教示され、元の命令編集画面に戻ります。" いいえ " に対応するファンクションキー([F1]) を押すと教示を中止することができます。



現在位置の登録 [F1]



以上で現在位置の教示操作は終了です。

◇◆◇ひとつの位置変数だけが対象◇◆◇

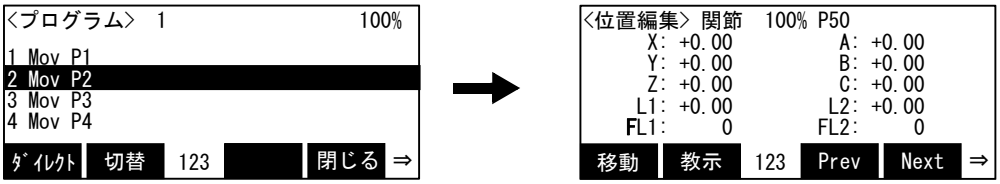
読み出したステップが、"Mov P1+P2" や "P1=P10" など、複数の位置変数を使っている場合は、最も左側の位置変数が教示の対象となります。また、"Mov p1+P2" のように大文字と小文字の位置変数が混在している場合は大文字の位置変数が教示の対象となります。
他の変数を教示する場合は、次頁の "(b) 位置編集画面での教示 " を参照願います。

(b) 位置編集画面での教示

以下に、位置変数 P5 へ現在位置を教示する場合の操作手順を示します。あらかじめジョグ操作などでロボットを作業位置へ移動させておきます。

1) 位置編集画面の表示

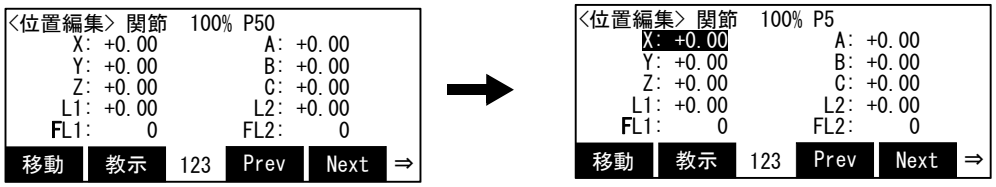
命令編集画面で、″切替″に対応するファンクションキー([F2])を押して位置編集画面を表示します。



位置編集の表示 [F2]

2) P5 の呼び出し。

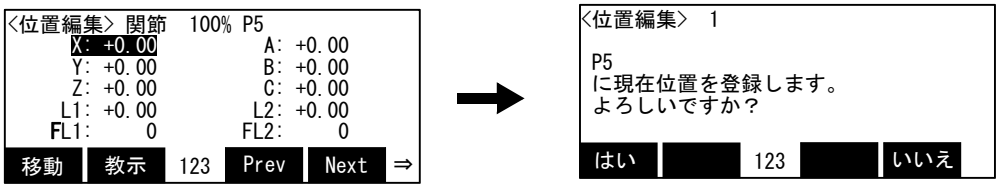
″Prev″、″Next″に対応するファンクションキー([F3]),([F4])を押して″P5″を呼び出します。



ステップ 5 の呼び出し [F3] [F4]

3) 現在位置の教示

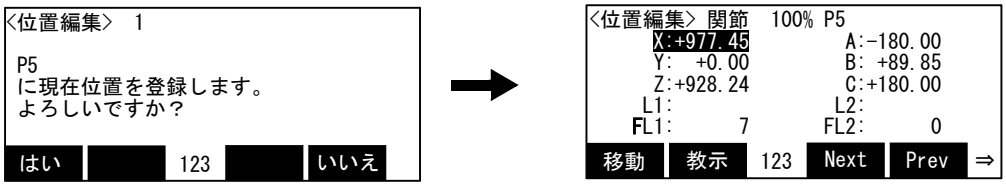
″教示″に対応するファンクションキー([F2])を押すと、確認画面を表示します。



ステップ 5 の呼び出し [F2]

4) ″はい″に対応するファンクションキーを押すと、ロボットの現在位置データが P5 に教示され、元の位置編集画面に戻ります。P5 にはロボットの現在位置データが表示されています。

″いいえ″に対応するファンクションキー([F1])を押すと教示を中止することができます。

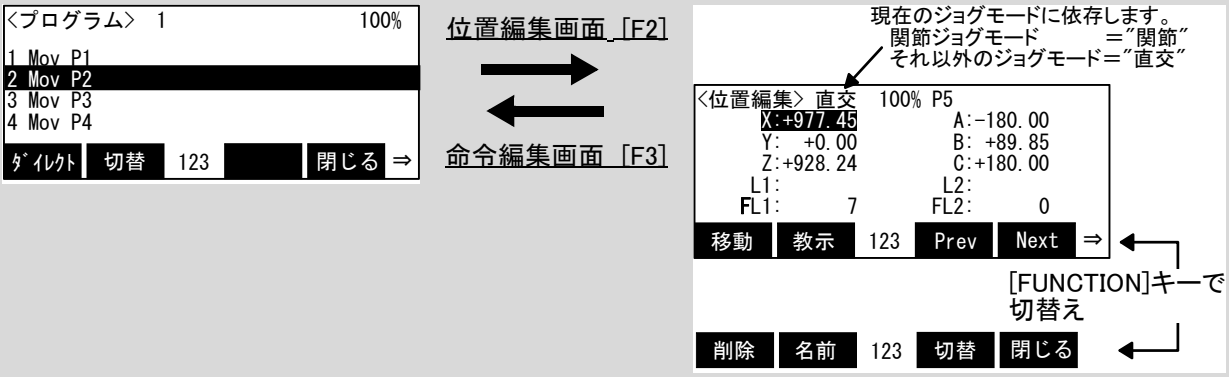


現在位置の登録 [F1]

以上で現在位置の教示操作は終了です。

◇◆◇命令編集画面と、位置編集画面の切り替え◇◆◇

″切替″に対応するファンクションキー(F2)を押すと、命令編集画面と、位置編集画面を相互に切り替えることができます。
画面に″切替″が表示されていない場合は、[FUNCTION]キーを押すと表示されます。メニューの右端に″⇒″が表示されている場合は、[FUNCTION]キーを押すことでメニューを切り替えられる状態を示しています。

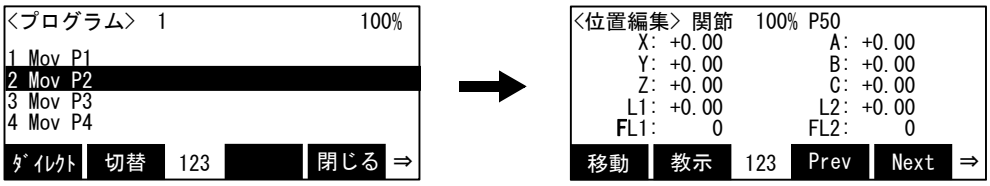


Prev(F3), Next(F4)で前後の位置変数を順次呼び出すことができます。通常は位置変数のみの呼び出しですが、ファンクションキーを切り替えて、名前(F2)で位置変数に加えて関節変数を呼び出しができます。関節変数を呼び出した後は、Prev(F3), Next(F4)で前後の関節変数を順次呼び出すことができます。
関節変数で先頭または最後まで表示すると、次の表示で位置変数に戻ります。

(6) 位置変数の削除

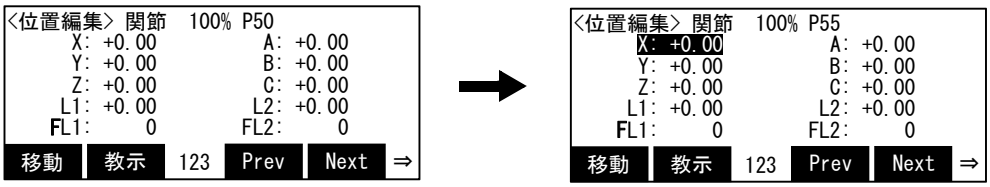
位置変数を削除する操作手順を示します。
プログラムで使用されていない変数に限り削除できます。

1) 位置編集画面の表示
命令編集画面で、″切替″に対応するファンクションキー([F2])を押して位置編集画面を表示します。



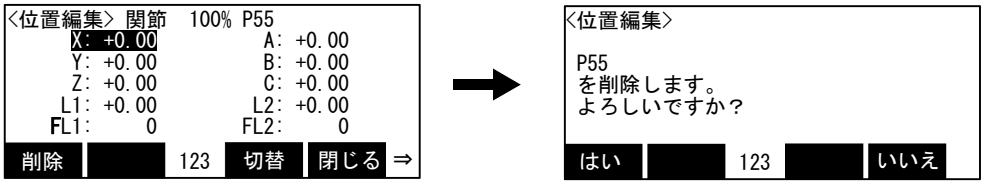
位置編集画面の表示 [F2]

2) 削除する位置変数の表示
″Prev″、または″Next″に対応するファンクションキー([F3]), または([F4])を押して削除する位置変数を表示します。



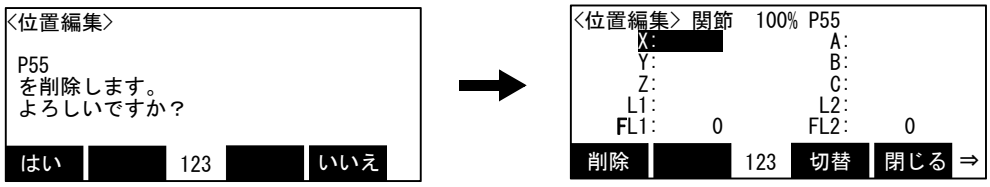
ステップ 5 の呼び出し [F3] [F4]　　P55 を削除する場合

3) 位置変数の削除
″削除″に対応するファンクションキー([F1])を押すと、確認画面を表示します。
(″削除″が表示されていない時は [FUNCTION] キーを押すと表示されます)



ステップ 5 の呼び出し [F1]　　現在位置の登録 [F1]

4) 位置変数の削除
″はい″に対応するファンクションキー([F1])を押すと、位置変数は削除されます。



ステップ 5 の呼び出し [F1]　　現在位置の登録 [F1]

(7) 位置データの確認

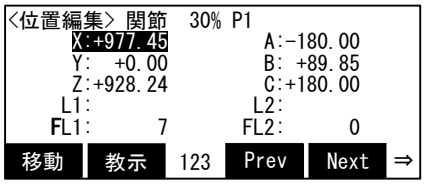
教示されている位置変数の場所へ、ロボットを移動させることができます。
移動モードには関節モードと、直交モードがあり、以下のように現在設定されているジョグモードに依存します。

表 3-4：指定された位置データへの移動方法

名称	移動方法
関節モード	画面の表示されている位置変数の場所まで、関節補間で移動します。 ジョグモードが関節モードの時に、この移動方法となります。 Mov 命令と同様な動きになります。
直交モード	指定された位置データの場所まで直線補間で移動します。従って現在位置と指定された位置の構造フラグが異なると移動できません。ジョグモードが直交、3 軸直交、円筒、およびツールジョグの時に、この移動方法となります。 Mvs 命令と同様な動きになります。

以下に操作方法を示します。
本操作はサーボ ON してイネーブルスイッチ(3 ポジションスイッチ)を軽く支えたまま、サーボ ON 状態を維持しておこないます。

1) 予め移動させたい位置変数を表示しておきます。
〝移動〟に対応するファンクションキーを押すと、キーを押し続けている間表示されている位置変数の場所へ向かってロボットが動作します。
〝移動〟に対応するファンクションキーを離すとロボットは停止します。また、イネーブルスイッチ(3 ポジションスイッチ)を離すか、さらに強く押さえるとサーボ OFF してロボットは停止します。



位置編集画面の表示 [F1]

⚠ 注意 本操作でロボットが動作します。
画面に表示されている位置変数の場所へロボットが動作しても、周辺装置などと干渉しないことを確認した上でおこなってください。
最初は速度を下げておくことをお勧めします。また、移動モード(関節、直交)によるロボットの動作軌跡を予測しながらおこなうことも大切です。

◇◆◇動作速度◇◆◇
[OVRD ↑]キー(速度アップ)、[OVRD ↓]キー(速度ダウン)を押すと移動速度を調整できます。

(8) 位置データの MDI(Manual Data Input) 登録 (修正)

MDI とは、位置データの各軸成分データに直接数値を入力して登録する方法です。
位置データ同士を加算して、ある基準位置からの相対移動量 (差分) として使用する位置変数の登録や、登録済みの位置データを微調整する場合などに有効な登録方法です。

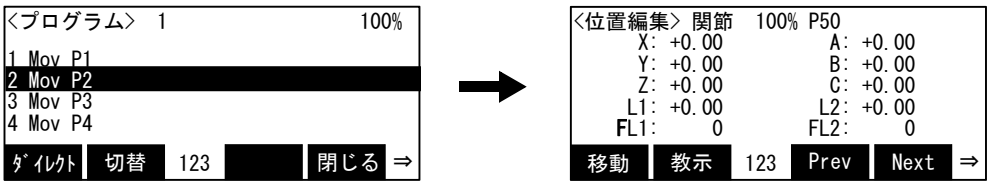
参考) 相対移動量としての位置データ

例)P1 を基準位置として、そこから 50mm 上空に関節補間で移動させる場合、P1 は現在位置の教示で登録、P50 は Z 軸成分に "50.00"、他の軸成分には "0.00" を MDI 登録した位置変数を使って Mov P1+P50 を実行することで可能です。

以下に、上記例の P50 を MDI で登録する場合の操作方法を示します。

1) 位置編集画面の表示

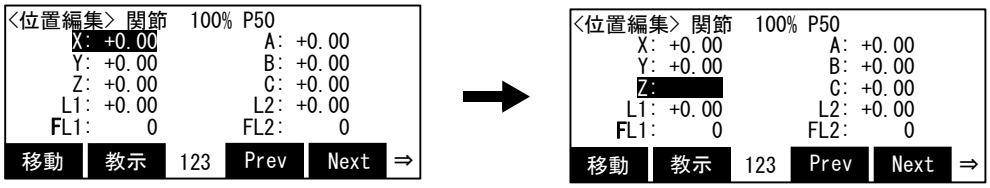
命令編集画面で、" 切替 " に対応するファンクションキーを押して位置編集画面を表示します。



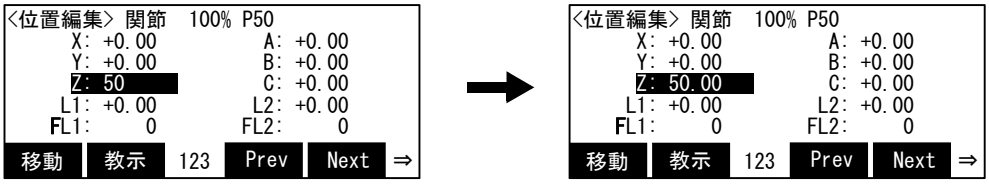
位置編集画面の表示 [F2]

2) Z 軸に "50" を入力する。

[↓] キーを 2 回押してカーソルを Z 軸へ移動させます。[CLEAR] キーを長押しして表示されている "+0.00" を削除します。[5]、[0]、[EXE] キーを押します。位置変数 P50 は、Z 軸の値のみが 50mm として登録されます。



カーソル移動と値のクリア [↓] [CLEAR]



値 50 の入力 [5] [0] [EXE]

3.6 デバッグ操作

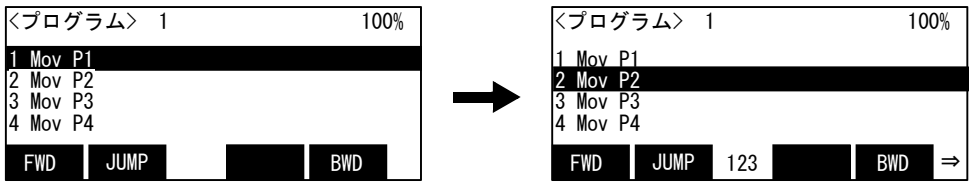
デバッグとは、作成したプログラムが正常に動作するかどうかのテストをおこなうと共に、異常時にはその誤りを修正する作業で、T/B のデバッグ機能を用いて確認することができます。実行可能なデバッグ機能は以下のとおりです。プログラムを作成した場合には必ずデバッグをおこない、プログラムに誤りがないことを確認してから運転をおこなってください。

(1) ステップ送り

T/B を使って、プログラムを 1 ステップづつ実行して (ステップ運転) 動作を確認します。
デバッグ操作を行う前に、34 ページの「3.5.1 プログラムの作成」 の 34 ページの「(1) プログラム編集画面のオープン」を参照してデバッグの対象となるプログラムの編集画面を表示させてください。
ロボットの動作命令ではサーボオン操作後、T/B のイネーブルスイッチ (3 ポジションスイッチ) を軽く押さえたままサーボ ON 状態を維持しておこなってください。

1) ステップ送りの実行

"FWD" に対応するファンクションキー ([F1]) を押すと、押し続けている間だけカーソルがあるステップを実行します。
ファンクションキーを途中で離すと、実行は中断します。また、ロボットの移動中はイネーブルスイッチ (3 ポジションスイッチ) を離すか、さらに強く押し込むことサーボ OFF し、実行を中断することができます。
実行中はオペレーティングパネルの [START] スイッチの LED が点灯します。1 ステップの実行を完了すると [START] スイッチの LED は消灯し、[STOP] スイッチの LED が点灯します。[F1] キーを離すと T/B 画面のカーソルは次のステップへ移動します。



ステップ送り [F1]

"FWD" に対応するファンクションキーを押す毎に次のステップへ進みます。

⚠ 注意 動作中はロボットの動きに十分注意してください。周辺装置との干渉など異常時には "FWD" に対応するファンクションキーを離すか、イネーブルスイッチ (3 ポジションスイッチ) を離すか強く押さえてロボットを停止させてください。

⚠ 注意 ロボットは、その時の指定速度によって異なる軌跡で動作する場合があります。
特に、コーナー部分は内回り距離が変化する場合がありますので、自動運転を始める時は最初、低速で作動させ、周辺装置との干渉に注意しながら徐々に速度を上げてください。

◇◆◇ステップ運転とは◇◆◇

プログラムを 1 ステップずつ実行することをステップ運転といいます。動作速度が遅く、1 ステップごとに実行が停止するので、プログラムの確認や動作位置の確認作業に有効です。
カーソルのあるステップが実行されます。
End 命令や Hlt 命令を実行すると、それ以上ステップ送りは進みません。

◇◆◇実行ステップの変更◇◆◇

矢印キーでのカーソル移動や、ジャンプ操作 ("JUMP") で実行ステップを変更できます。

◇◆◇動作中のロボットを直ちに停止するには◇◆◇

① [EMG. STOP] スイッチを押す。
サーボ OFF し、非常停止エラーが発生して動作中のロボットは直ちに停止します。
継続は、エラーリセット → サーボ ON → ステップ運転操作の順におこないます。

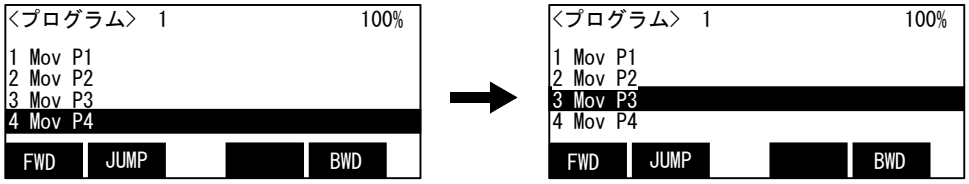
②イネーブルスイッチ (3 ポジションスイッチ) を離す、またはさらに強く押し込む。
サーボ OFF し、エラー 2000 が発生して動作中のロボットは直ちに停止します。
継続は、エラーリセット → サーボ ON → ステップ運転操作の順におこないます。

(2) ステップ戻し

ステップ送りや自動運転で実行の終わったステップ、あるいは中断しているステップを、補間命令に限り 1 ステップずつ戻し方向に実行します。なお、戻すことができるのは 4 ステップまでです。

1) ステップ戻しの実行

″BWD″ に対応するファンクションキーを押すと、押し続けている間だけステップの戻り方向へ 1 ステップだけ実行します。
ファンクションキーを途中で離すと、実行は中断します。また、ロボットの移動中はイネーブルスイッチ (3 ポジションスイッチ) を離すか、さらに強く押し込むことサーボ OFF し、実行を中断することができます。
実行中はオペレーティングパネルの [START] スイッチの LED が点灯します。1 ステップの実行を完了すると [START] スイッチの LED は消灯し、[STOP] スイッチの LED が点灯します。T/B 画面のカーソルはステップの戻り方向で次の補間命令のステップへ移動します。

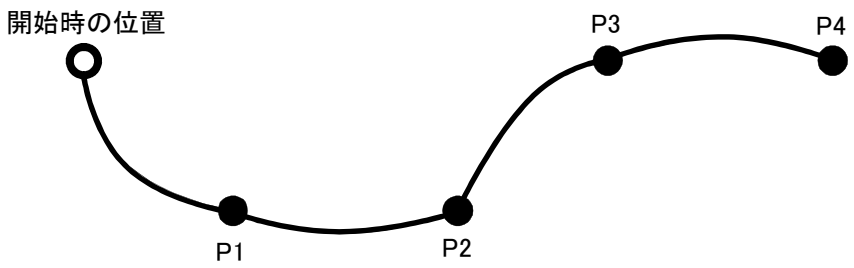


ステップ戻し [F4]

″BWD″ に対応するファンクションキーを押す毎に前のステップへ戻ります。

[補足] 下記プログラムをステップ 4 までステップ送りした後にステップ戻しをおこなう場合、P1 まで戻った後にさらにステップ戻しをおこなうと、ステップ送りをおこなった開始時の位置まで戻ります。(開始時の位置とはステップ 1 の実行を始めた位置です)

プログラム
1 Mov P1
2 Mov P2
3 Mov P3
4 Mov P4
:
:



⚠ 注意 動作中はロボットの動きに十分注意してください。周辺装置との干渉など異常時には ″BWD″ に対応するファンクションキーを離すか、イネーブルスイッチ (3 ポジションスイッチ) を離すか強く押さえてロボットを停止させてください。

◇◆◇動作中のロボットを直ちに停止するには◇◆◇

① [EMG. STOP] スイッチを押す。
サーボ OFF し、非常停止エラーが発生して動作中のロボットは直ちに停止します。
継続は、エラーリセット → サーボ ON → ステップ運転操作の順におこないます。

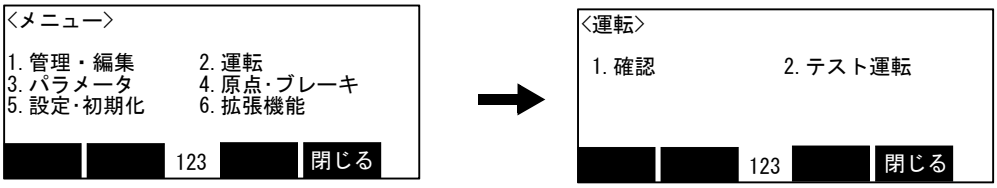
②イネーブルスイッチ (3 ポジションスイッチ) を離す、またはさらに強く押し込む。
サーボ OFF し、エラー 2000 が発生して動作中のロボットは直ちに停止します。
継続は、エラーリセット → サーボ ON → ステップ運転操作の順におこないます。

(3) 別スロットでのステップ送り

マルチタスクのプログラムを確認する場合は運転メニューの確認画面でステップ送りができます。

1) 運転メニューの選択

メニュー画面で [2] キーを押して "2. 運転 " を選択します。

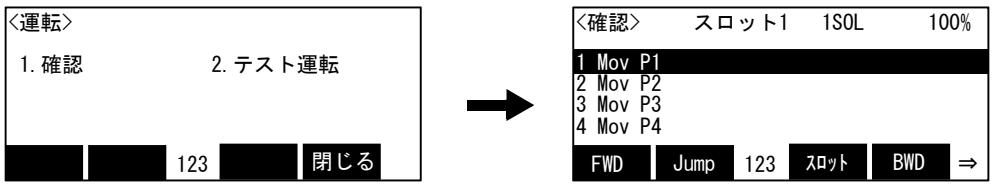


運転画面の選択 [2]

2) 確認画面の選択

運転画面で [1] キーを押して "1. 確認 " を選択します。

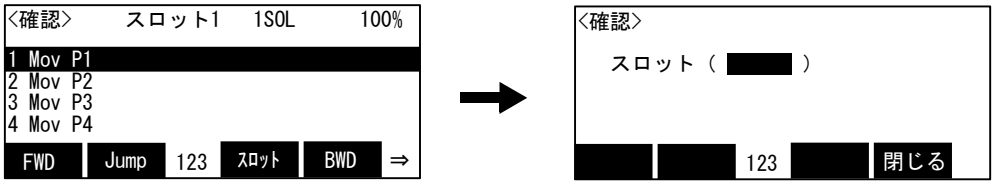
スロット 1 に設定されているプログラムを表示します。スロット番号に続いてプログラム名称が表示されます。



運転画面の選択 [1]

3) スロットの変更

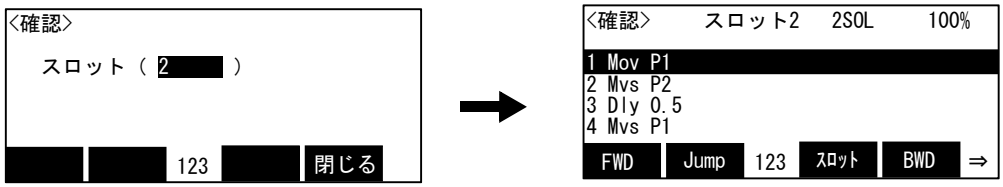
" ｽﾛｯﾄ " に対応するファンクションキー ([F3]) を押すと、スロット番号指定画面を表示します。



スロットの変更 [F3]

希望するスロット番号を入力して [EXE] キーを押します。

入力したスロット番号のプログラムを表示します。(下記はスロット 2 を指定した場合)



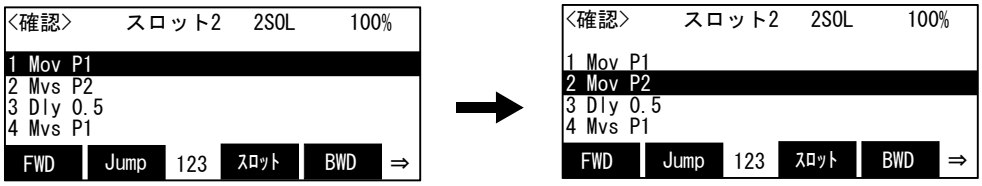
スロットの変更 [2] [EXE]

4) ステップ運転の実行

ステップ送りと、ステップ戻しは命令編集画面でのステップ運転操作と同様に実行できます。

"FWD" に対応するファンクションキー ([F1]) を押すとステップ送り、"BWD" に対応するファンクションキー ([F4]) を押すとステップ戻しを、それぞれのファンクションキーを押し続けている間だけカーソルがあるステップを実行します。

ファンクションキーを途中で離すと、実行は中断します。また、ロボットの移動中はイネーブルスイッチ(3 ポジションスイッチ)を離すか、さらに強く押し込むことでサーボ OFF し、実行を中断することができます。
実行中はオペレーティングパネルの [START] スイッチの LED が点灯します。1 ステップの実行を完了すると [START] スイッチの LED は消灯し、[STOP] スイッチの LED が点灯します。T/B 画面のカーソルは次のステップへ移動します。



ステップ送り [F1]　（ステップ戻し [E4]）

⚠ 注意

動作中はロボットの動きに十分注意してください。周辺装置との干渉など異常時には "FWD" または "BWD" に対応するファンクションキーを離すか、イネーブルスイッチ(3 ポジションスイッチ)を離す、またはさらに強く押さえてロボットを停止させてください。

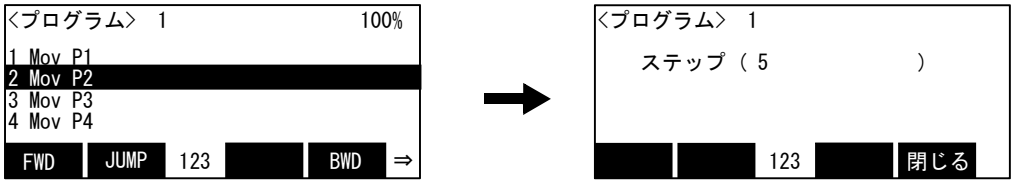
◇◆◇実行ステップの変更◇◆◇

矢印キーでのカーソル移動や、ジャンプ操作 ("JUMP") で実行ステップを変更できます。
後述の 51 ページの「(4) ステップジャンプ」を参照してください。

(4) ステップジャンプ

ステップ運転をおこなうステップ番号を変更することができます。
以下に例としてステップ 5 からステップ運転をおこなう場合の操作を示します。

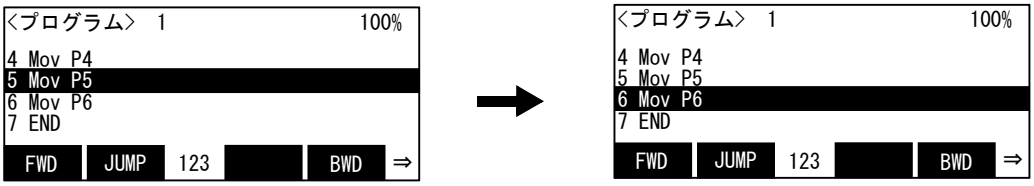
1) ステップ 5 の呼び出し。
″JUMP″ に対応するファンクションキー ([F2]) を押して表示されるステップ番号入力画面で、[5]、[EXE] キーを押します。ステップ 5 にカーソルが移動します。



ステップ 5 の呼び出し [F2]　　ステップ 5 の呼び出し [5] [EXE]

[↑]、[↓] キーでステップ 5 にカーソルを合わせてもステップ 5 を呼び出すことができます。

2) ステップ送り
前述の 46 ページの「(1) ステップ送り」 と同様に、″FWD″ に対応するファンクションキー ([F1]) を押すと、ステップ 5 からステップ送りをおこなうことができます。



ステップ送り [F1]

⚠ 注意　実行するステップを変更してステップ運転をおこなう場合は、ロボットが周辺装置などと干渉しないことを予め確認した上でおこなってください。
また、動作中はロボットの動きに十分注意し、周辺装置との干渉など異常時には ″FWD″ または ″BWD″ に対応するファンクションキーを離すか、イネーブルスイッチ (3 ポジションスイッチ) を離す、またはさらに強く押さえてロボットを停止させてください。

3.7 自動運転

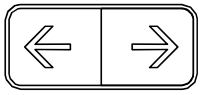
3.7.1 動作速度の設定

ロボットの動作速度の設定は、コントローラの O/P(操作パネル)、または T/B(ティーチングボックス) でおこないます。自動運転時の実際の動作速度は、
動作速度 =(コントローラまたは T/B の設定値)×(プログラムの設定値) になります。

(1) O/P での操作

オーバーライドの表示

1) コントローラの [CHNG DISP] スイッチを 2 回押して、STATUS NUMBER 表示パネルに ″ オーバーライド ″ を表示させます。



オーバーライドの設定

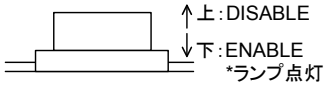
2) [UP] ボタンを押す毎にオーバーライドが (10 → 20 → 30 → 40 → 50 → 60 → 70 → 80 → 90 → 100%) の順に上がります。
[DOWN] ボタンを押すと逆に下がります。

(2) T/B での操作

1) [OVRD ↑] キーを押す毎にオーバーライドが (3 → 5 → 10 → 30 → 50 → 70 → 100%) の順に上がります。[OVRD ↓] キーを押すと逆に下がります。

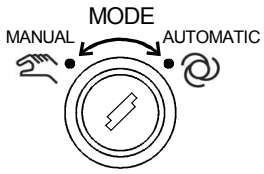
3.7.2 プログラムの選択

コントローラの準備



T/B 無効　　T／B背面

1) T/B の [ENABLE] スイッチを ″DISABLE″ にしてください。



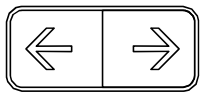
コントローラ有効

2) コントローラのモードを ″AUTOMATIC″ にします。

プログラム番号の選択

プログラム番号の表示

3) [CHNG DISP] ボタンを押して、STATUS NUMBER 表示パネルに
″ プログラム番号 ″ を表示させます。
(先頭に ″P″ が表示されます。)



プログラム番号の選択

4) [UP] ボタンを押すと登録プログラムを順にアップ表示、[DOWN] ボタンを押すと逆にダウン表示します。
自動運転の対象となるプログラム番号を表示させてください。
※ プログラム名が 5 文字以上の場合は表示されません。これらを外部機器から選択した場合は ″P-----″ を表示します。

◇◆◇ T/B でのプログラム選択◇◆◇
T/B のソフトウェアバージョンが 1.3 以降では、T/B からの操作でプログラム選択ができます。
操作方法の詳細は 67 ページの「(6) プログラムの選択」 を参照願います。

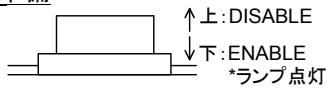
3.7.3 自動運転の開始

(1) O/P からの起動

⚠注意 自動運転をおこなう前に、必ず以下の項目を確認してください。確認をせずに自動運転をおこなうと、物損や人身事故の原因となります。

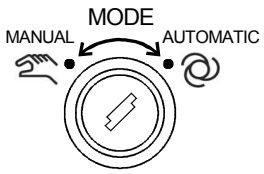
- ・ロボットの周りに人がいないこと。
- ・安全柵がロックされていて、不用意に人が近づけないこと。
- ・ロボットの動作範囲内に工具など不要なものが置いてないこと。
- ・ワークは所定の位置に正しく置いてあること。
- ・ステップ運転でプログラムが正しく動作することが確認されていること。

コントローラの準備



T/B 無効　　T／B背面

1) T/B の [ENABLE] スイッチが ″DISABLE″ になっていることを確認します。



コントローラ有効

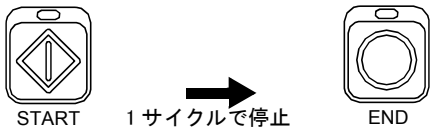
2) コントローラのモードが ″AUTOMATIC″ になっていることを確認します。

サーボオン

サーボオン

3) コントローラの [SVO ON] ボタンを押して、サーボ電源をオンにします。SVO ON ランプが点灯します。

自動運転の開始



開始(繰返し運転)

4) コントローラの [START] ボタンを押すと自動運転を開始します。(連続運転)
連続運転の状態で [END] ボタンを押すと 1 サイクルで停止します。サイクル停止中は LED が点滅します。サイクル停止中に再度 [END] ボタンを押すと連続運転に戻ります。

⚠注意 自動運転を開始するときは、必ず目的とするプログラム番号が選択されていることを確認してください。

⚠注意 自動運転中はロボットの動きに十分注意してください。
異常を発見したときは、[EMG. STOP] スイッチを押して直ちにロボットを停止させてください。

⚠注意 自動運転を開始する際は、はじめに低速状態で動作させ、周辺装置との干渉に注意しながら、徐々に速度を上げてください。

(2) T/B からの起動

T/B ソフトウェア Ver. 1.7 版以降では、T/B からの操作でプログラムの自動運転を開始することができます。(R57TB は Ver. 3.0 以降)

操作は、< メニュー > → < 運転 > から選択できる < 操作パネル > 画面でおこないます。

本機能は、パラメータ：TBOP の設定で無効にすることもできます。

⚠注意 ″AUTOMATIC″ モード時に T/B で < 操作パネル > 画面を表示している間は、操作権は T/B に移ります。(T/B の状態表示ランプ [ENABLE] が点滅)
この時は、コントローラの O/P (操作パネル)、および外部信号による操作権を必要とする操作はできませんのでご注意願います。(13 ページの「2.3.1 操作権について」を参照)

以下に、T/B の < 操作パネル > 画面から実行できる機能、および < 操作パネル > 画面を表示している時に実行できる操作を示します。

< 操作パネル > 画面からの操作 (各ファンクションキー)

①プログラムの選択 ..... ″ 選択 ″
②サーボオン / オフ ..... ″ サーボオン ″ / ″ サーボオフ ″
(コントローラのモードが ″AUTOMATIC″ の時は、イネーブルスイッチ ON は不要です)
③自動運転の実行 ..... ″ 起動 ″
④運転モード (連続 / サイクル) の変更..... ″ 連続 ″ / ″ サイクル ″
⑤プログラムリセット ..... ″ リセット ″

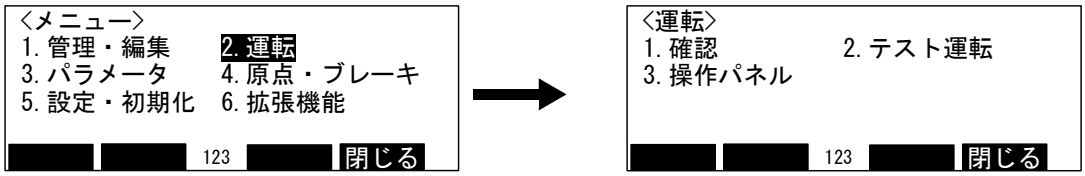
その他のキー操作

①動作速度の変更 .....[OVRD ↑] / [OVRD ↓] キー
注) コントローラのモードが ″AUTOMATIC″ に設定されていると通常の速度で動作、″MANUAL″ に設定されていると低速で動作 (動作確認用) します。
コントローラを希望するモードに切り替えてください。
②サーボオン .....[SERVO] キー
(コントローラのモードが ″AUTOMATIC″ の時は、イネーブルスイッチ ON は不要です)
③アラームリセット .....[RESET] キー

※[EMG. STOP] スイッチ、および [STOP] キーによるロボットの停止は常に有効です。
[JOG] 、[HAND]、[MONITOR] キーによる操作は無効となります。

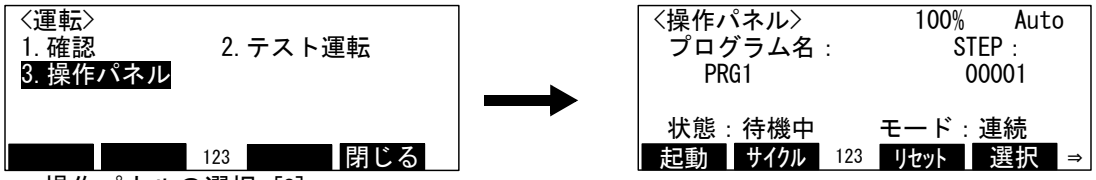
以下に、T/B から自動運転をおこなう時の操作方法を説明します。コントローラのモードを ″MANUAL″ に設定して自動運転 (確認運転) をおこなう場合は、T/B の [TB ENABLE] スイッチを押して T/B を有効状態にしてください。

1) メニュー画面で [2] キーを押すと、運転メニュー画面を表示します。



運転メニューの選択 [2]

2) 運転メニュー画面で [3] キーを押すと、< 操作パネル > 画面を表示します。



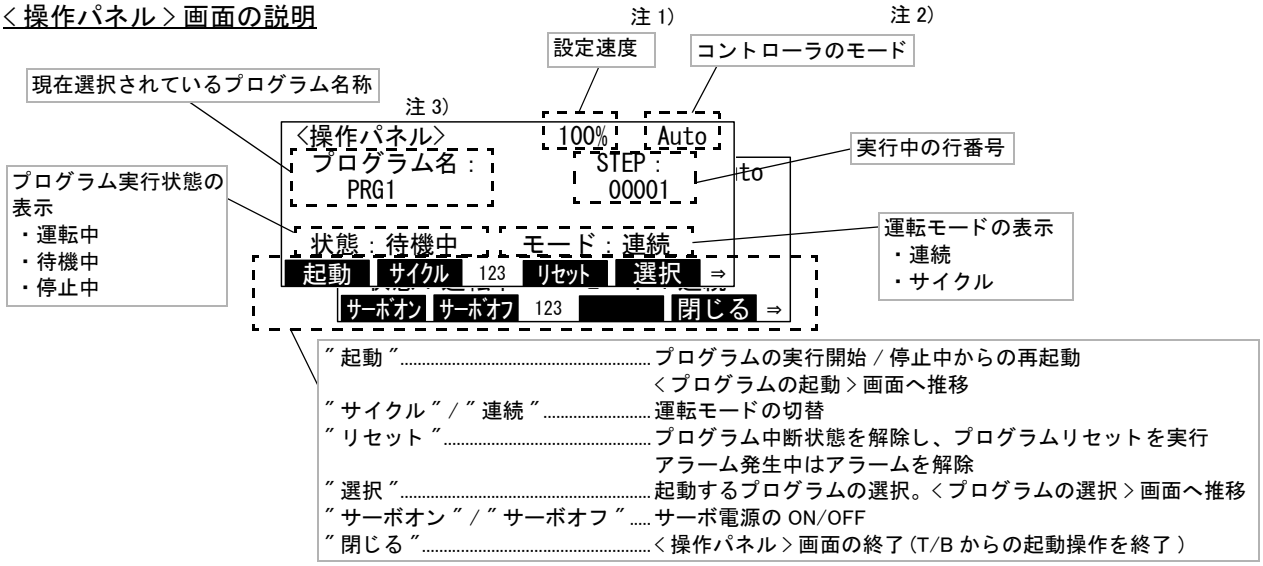
操作パネルの選択 [3]

⚠注意

自動運転をおこなう前に、必ず以下の項目を確認してください。確認をせずに自動運転をおこなうと、物損や人身事故の原因となります。

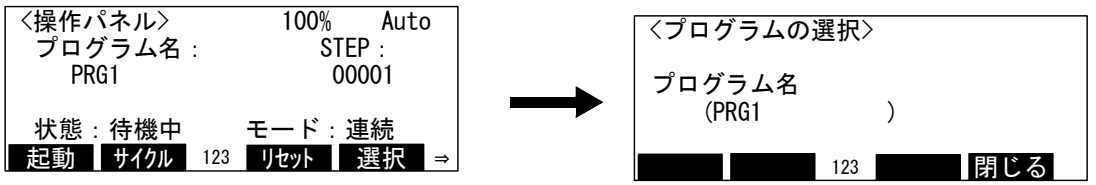
- ロボットの周りに人がいないこと。
- 安全柵がロックされていて、不用意に人が近づけないこと。
- ロボットの動作範囲内に工具など不要なものが置いてないこと。
- ワークが所定の位置に正しく置いてあること。
- ステップ運転でプログラムが正しく動作することが確認されていること。

<操作パネル>画面の説明



注 1) コントローラのモードが ″MANUAL″ の時は、設定速度が 100% でも低速動作となります。(動作の確認用)
注 2) コントローラのモードが ″AUTOMATIC″ の時は、T/B の状態表示ランプ [ENABLE] が点滅し、T/B が有効状態であることを示します。<操作パネル>画面を終了すると T/B は無効状態に戻ります。
注 3) <操作パネル>画面表示中は、[JOG] 、[HAND]、[MONITOR] キーは無効となります。

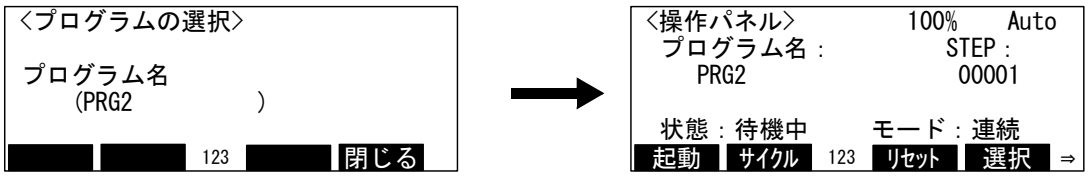
3) 操作パネル画面で ″選択″ に割り当てられているファンクションキー [F4] を押すと、プログラムの選択画面を表示します。



プログラムの選択 [F4]

現在選択されているプログラム名称を表示します。

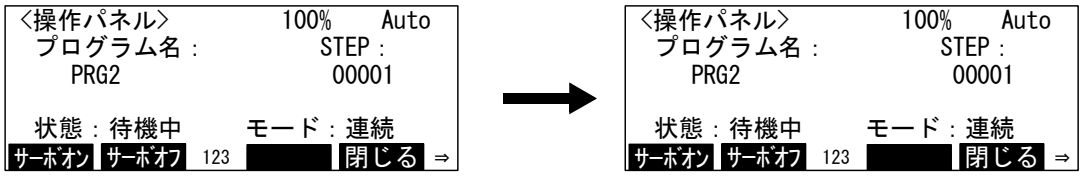
4) プログラム名のカッコ内に起動するプログラム名称を入力して、[EXE] キーを押します。
新しくプログラムを選択して、<操作パネル>画面に戻ります。



プログラム名称の入力、[EXE] キー

図はプログラム名を ″PRG2″ に変更した例です。

5)〝サーボオン〟に割り当てられているファンクションキー[F1]を押してサーボ電源を ON させてください。(〝サーボオン〟のファンクションが表示されていない時は[FUNCTION]キーを押します)



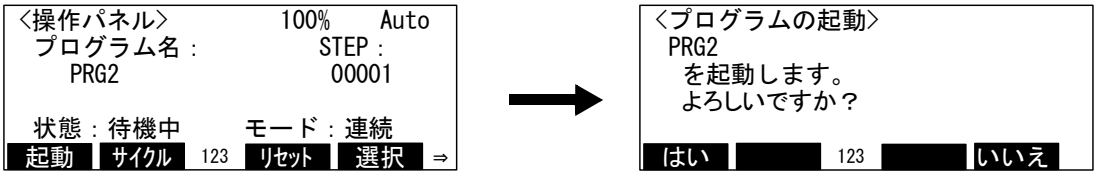
サーボオン[F1]

⚠注意 自動運転を開始するときは、必ず目的とするプログラム番号が選択されていることを確認してください。

⚠注意 自動運転中はロボットの動きに十分注意してください。
異常を発見したときは、[EMG. STOP]スイッチを押して直ちにロボットを停止させてください。

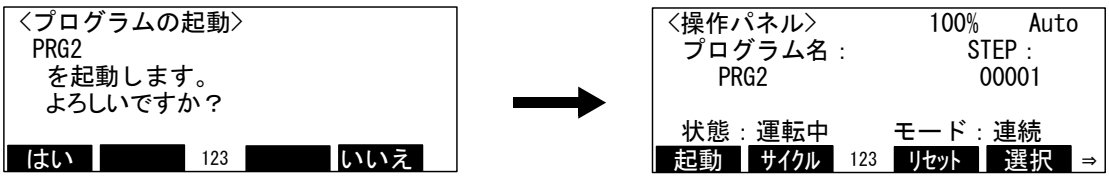
⚠注意 自動運転を開始する際は、はじめに低速状態で動作させ、周辺装置との干渉に注意しながら、徐々に速度を上げてください。T/B の[OVRD ↑][OVRD ↓]キーで動作速度を変更することができます。

6)〝起動〟に割り当てられているファンクションキー[F1]を押すと、確認画面を表示します。



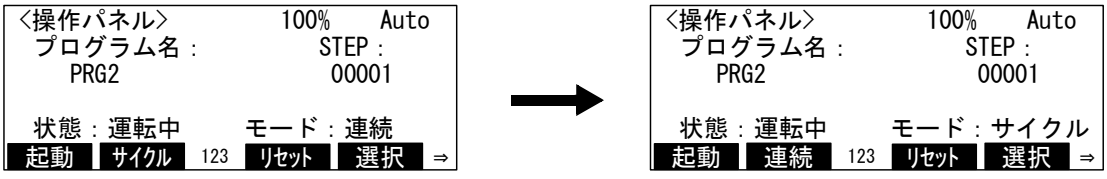
起動[F1]

7)〝はい〟に割り当てられているファンクションキー[F1]を押すと、現在選択されているプログラムの自動運転を開始します。
画面は〈操作パネル〉画面に戻ります。



はい[F1]

8) 運転モードは画面に表示されているモードに基づきます。
必要に応じて変更してください。



運転モードの変更[F2]

図は〝連続〟から〝サイクル〟に切り替えた例です。

以上で T/B からの自動運転の開始操作は終了です。

3.7.4 停止

運転中のプログラムを直ちに停止させ、移動中のロボットを減速停止させます。

(1) O/P での操作

停止　STOP

1) [STOP] ボタンを押します。

(2) T/B での操作

停止　[STOP]

1) [STOP] キーを押します。

◇◆◇停止操作は操作権を問いません◇◆◇

停止操作は、操作権の有無にかかわらず常に有効です。

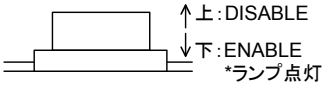
3.7.5 停止からの自動運転再開

(1) O/P からの再起動

⚠注意 自動運転をおこなう前に、必ず以下の項目を確認してください。確認をせずに自動運転をおこなうと、物損や人身事故の原因となります。

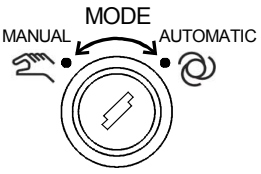
- ロボットの周りに人がいないこと。
- 安全柵がロックされていて、不用意に人が近づけないこと。
- ロボットの動作範囲内に工具など不要なものが置いてないこと。
- ワークは所定の位置に正しく置いてあること。
- ステップ運転でプログラムが正しく動作することが確認されていること。

コントローラの準備



T/B 無効　　T／B背面

1) T/B の [ENABLE] スイッチを ″DISABLE″ にしてください。



コントローラ有効

2) コントローラのモードを ″AUTOMATIC″ にします。

自動運転の再開

開始(繰返し運転)

3) コントローラの [START] ボタンを押すと自動運転を再開します。(繰返し運転)
繰返し運転 /1 サイクル運転は前の状態を保持します。

⚠注意 自動運転を再開するときは、必ず目的とするプログラム番号が選択されていることを確認してください。

⚠注意 自動運転中はロボットの動きに十分注意してください。
異常を発見したときは、[EMG. STOP] スイッチを押して直ちにロボットを停止させてください。

⚠注意 自動運転中にはロボット CPU の電源を切らないでください。ロボット CPU 内のメモリが異常になりプログラムが壊れる可能性があります。ロボットを即座に停止させる場合は非常停止をご利用ください。

(2) T/B からの再起動

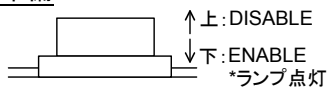
操作方法は自動運転の開始と同じです。53 ページの「3.7.3 自動運転の開始」を参照して T/B からの再起動操作をおこなってください。

3.7.6 プログラムリセット

プログラムの中断状態を解除し、実行行を先頭に戻します。

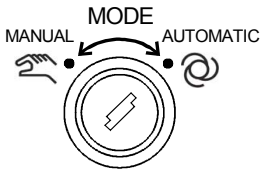
(1) O/P での操作

コントローラの準備



T/B 無効　　T／B背面

1) T/B の [ENABLE] スイッチを ″DISABLE″ にしてください。



コントローラ有効

2) コントローラのモードを ″AUTOMATIC″ にします。

プログラム番号の表示　CHNG DISP

3) コントローラの [CHNG DISP] ボタンを押して、″プログラム番号″を表示させます。

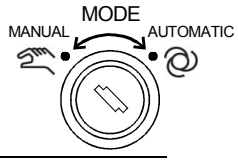
プログラムリセットの実行

リセット　RESET

4) コントローラの [RESET] ボタンを押します。STOP ランプが消灯し、プログラムの中断状態を解除します。

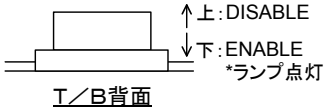
(2) T/B での操作

T/B の準備



コントローラ無効

1) コントローラのモードを ″MANUAL″ にします。



T/B 有効　　T／B背面

2) T/B の [ENABLE] スイッチを ″ENABLE″ にします。

プログラムリセットの実行

3) [RESET] キーを押したまま [EXE] キーを押してください。実行行が先頭に戻り、プログラムがリセットされます。

プログラムリセット　[RESET] + [EXE]

◇◆◇プログラムリセットはプログラムの中断中でのみ有効です◇◆◇

プログラム運転中は、プログラムのリセット操作はできません。中断中に操作してください。
また、コントローラの操作パネルからプログラムリセットをおこなう場合は、STATUS NUMBER 表示に ″プログラム番号″を表示させてからリセット操作をしてください。

◇◆◇ STOP ランプが消灯します◇◆◇

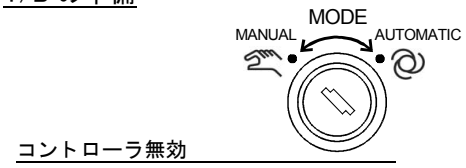
プログラムリセットがおこなわれると、STOP ランプが消灯します。

3.8 サーボオン/オフ

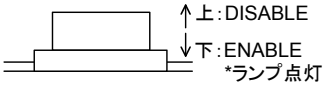
サーボ電源は、ティーチモード中は安全のため T/B の裏にあるイネーブルスイッチ(3 ポジションスイッチ)を軽く押している間のみオンします。T/B での操作はイネーブルスイッチ(3 ポジションスイッチ)を軽く押した状態でおこなってください。

■ T/B での操作

T/B の準備



コントローラ無効



T/B 有効　　T／B背面

1) コントローラのモードを "MANUAL" にします。

2) T/B の [ENABLE] スイッチを "ENABLE" にしてください。

3) [SERVO] キーを押すとサーボオンします。

4) イネーブルスイッチ(3 ポジションスイッチ)を離すか、さらに強く押し込むとサーボオフします。

サーボオンの実行

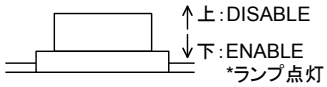
サーボ ON 操作　[SERVO]

サーボオフの実行

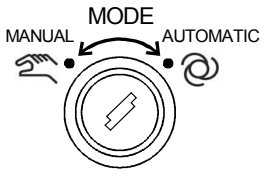
サーボ OFF 操作　イネーブルスイッチ(3 ポジションスイッチ)
離すか、強く押し込む

■ O/P での操作

コントローラの準備



T/B 無効　　T／B背面



コントローラ有効

1) T/B の [ENABLE] スイッチを "DISABLE" にしてください。

2) コントローラのモードを "AUTOMATIC" にします。

サーボオンの実行

サーボオン　SVO ON

3) [SVO. ON] ボタン押すとサーボオンし、SVO ON ランプが点灯します。

サーボオフの実行

サーボオフ　SVO OFF

4) [SVO. OFF] ボタン押すとサーボオフし、SVO OFF ランプが点灯します。

◇◆◇ブレーキが動作します◇◆◇
サーボオフすると、自動的にブレーキが動作します。ロボットの機種によってはブレーキのない軸もあります。

3.9 エラーリセット操作

■ O/P からのエラーリセット操作

エラーを解除

エラーリセット　RESET

1) コントローラの [RESET] ボタンを押します。
T/B 側のエラーがリセットされない場合は T/B からのリセット操作をおこないます。

■ T/B からのエラーリセット操作

エラーを解除

エラーリセット [RESET]

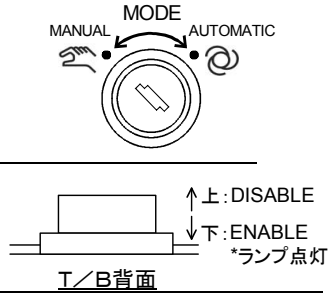
1) T/B の [RESET] キーを押します。

3.10 解除できないエラーの一時的エラーリセット操作

ロボットの機種によっては軸が動作範囲外にいる場合などは解除できないエラーが発生することがあります。この場合は通常の操作ではサーボオンとジョグ操作ができません。その場合は下記手順にて一時的にエラーを解除することができます。動作範囲外の場合などは一時的にエラーを解除している間にジョグ操作を行って動作範囲内に移動させてください。
また、干渉回避機能を使用中に、サーボ OFF 状態で人手によりロボットを干渉領域に入れてしまった場合なども、[RESET] キーを押したままでジョグ操作をおこなうことで、干渉回避機能を解除してロボットを動作させることができます。
なお [RESET] キーは、R56/57TB における [CAUTION] キーに相当します。R56/57TB をご使用の場合は、[CAUTION] キーを使用して下記の操作をおこなってください。

■ T/B からの一時的エラー解除操作

T/B の準備



コントローラ無効

T/B 有効　　T／B背面

1) コントローラのモードを "MANUAL" にします。

T/B の [ENABLE] スイッチを "ENABLE" にしてください。

エラーを解除 [SERVO] + [RESET]

2) イネーブルスイッチ (3 ポジションスイッチ) を軽く握り、[SERVO] キーを押しながら [RESET] キーを押し続けます。

以上の操作により一時的にエラーをリセットできます。
キーを離すと再度エラーが発生しますのでキーを離さないでください。
ジョグ操作をおこなう場合も [RESET] キーを押したまま操作してください。

注）この [RESET] キーを押したままでのジョグ操作では、動作範囲リミットで停止しません。動作範囲外の方向へ動かさないようご注意願います。

3.11 管理・編集画面の操作

ここでは、プログラム管理に関する以下の操作方法を説明します。

(1) プログラムの一覧表示
(2) プログラムのコピー (コピー)
(3) プログラムの名称変更 (リネーム)
(4) プログラムの削除 (削除)
(5) プログラムの保護 (プロテクト)
(6) プログラムの選択

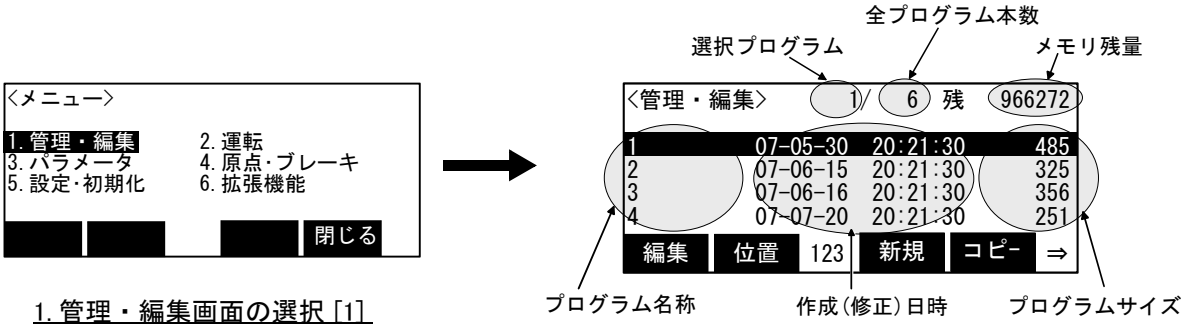
プログラムの編集方法については、34 ページの「3.5 プログラミング」を参照願います。

(1) プログラムの一覧表示

ロボット CPU 内のプログラム登録状態を表示する画面です。

1) 管理・編集メニューの選択。

メニュー画面で、[1] キーを押します。″1. 管理･編集 ″ が選択され、プログラムの一覧を表示します。



1. 管理・編集画面の選択 [1]

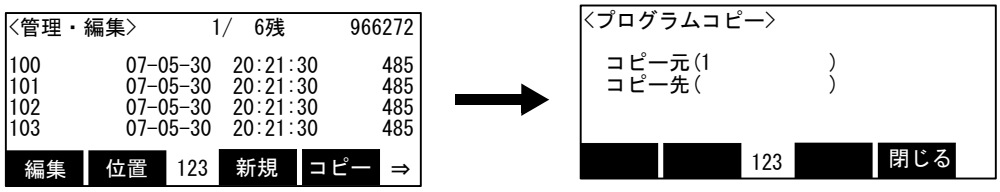
[↑]、[↓] キーで ″1. 管理・編集 ″ にカーソルを合わせて [EXE] キーを押しても同様の操作ができます。
プログラム一覧画面で [↑]、[↓] キーを押すと、他のプログラムを表示します。
また、各操作の対象となるプログラムを選択しておくこともできます。

画面の下にファンクションキーに対応するメニュー (″ 編集 ″、″ 位置 ″、″ 新規 ″、″ コピー ″) を表示しています。この内、″ 編集 ″、″ 位置 ″、″ 新規 ″ はプログラムの編集機能です。これらの操作方法については 34 ページの「3.5 プログラミング」を参照願います。
[FUNCTION] キーを押すと、″ リネーム ″、″ 削除 ″、″ プロテクト、″ 閉じる ″ を表示します。

(2) プログラムのコピー (コピー)

1) コピーメニューの選択

プログラム一覧表示で、″ コピー ″ に対応するファンクションキー ([F4]) を押します。コピー画面を表示します。

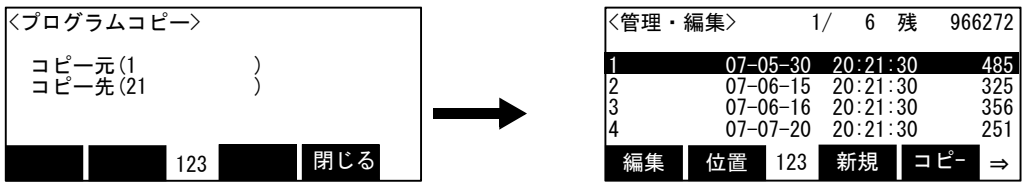


コピーの選択 [F4]

2) コピーするプログラムの指定と実行

コピー元のカッコ内には、予めプログラム一覧表示で選択されていたプログラム名称が表示されています。(図ではプログラム名称 ″1″) 変更する場合は、矢印キーでカーソルを移動して変更できます。

コピー先のカッコ内にコピーされるプログラム名称を入力し、[EXE] キーを押します。



プログラムの名称 [2] [1] [EXE]

この場合、プログラム ″1″ から ″21″ へコピーします。

◇◆◇保護 (プロテクト) 設定はコピーされません◇◆◇

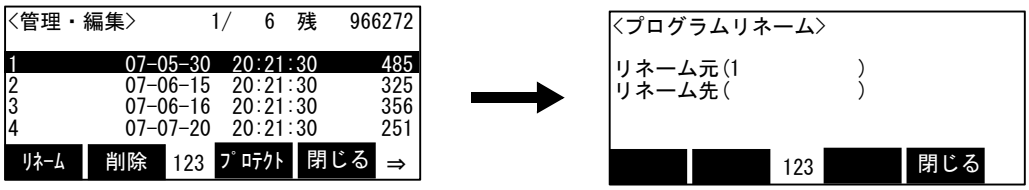
コピー操作では、プログラム保護情報、および変数保護情報はコピーされません。

必要に応じて再度設定してください。保護 (プロテクト) の設定方法は 66 ページの「(5) プログラムの保護 (プロテクト)」を参照してください。

(3) プログラムの名称変更 (リネーム)

1) リネームメニューの選択

プログラム一覧表示で、″ リネーム ″ に対応するファンクションキー ([F1]) を押します。コピー画面を表示します。″ リネーム ″ メニューが表示されていない場合は、[FUNCTION] キーを押して表示させます。

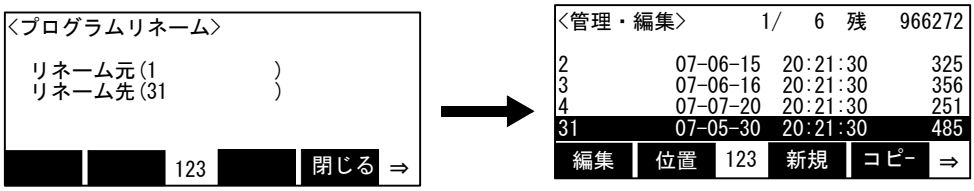


リネームの選択 [F1]

2) 名称を変更するプログラムの指定

リネーム元のカッコ内には、予めプログラム一覧表示で選択されていたプログラム名称が表示されています。(図ではプログラム名称 ″1″) 変更する場合は、矢印キーでカーソルを移動して変更できます。

リネーム先のカッコ内に新しいプログラム名称を入力し、[EXE] キーを押します。



この場合、プログラム名称 ″1″ を ″31″ へ変更します。

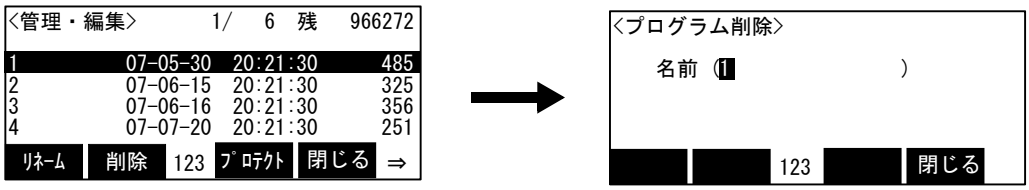
◇◆◇保護 (プロテクト) されているプログラム名称は変更できません◇◆◇

命令プロテクトが設定されているプログラム名称は変更できません。命令プロテクトを解除してから実行してください。命令プロテクトの解除方法は 66 ページの「(5) プログラムの保護 (プロテクト)」を参照してください。

(4) プログラムの削除 (削除)

1) 削除メニューの選択

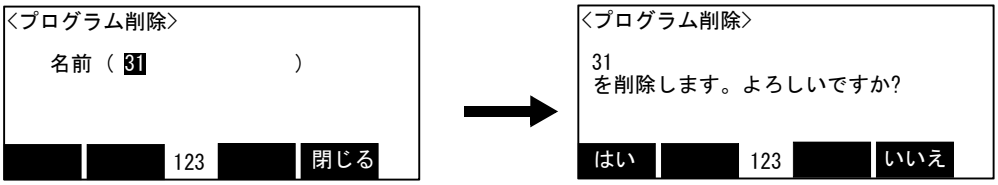
プログラム一覧表示で、″ 削除 ″ に対応するファンクションキー ([F2]) を押します。削除画面を表示します。″ 削除 ″ メニューが表示されていない場合は、[FUNCTION] キーを押して表示させます。



削除の選択 [F2]

2) 削除するプログラムの指定

名前のカッコ内には、予めプログラム一覧表示で選択されていたプログラム名称が表示されています。(図ではプログラム名称 ″1″) 変更する場合は、正しいプログラム名称を入力します。
[EXE] キーを押すと確認画面を表示します。

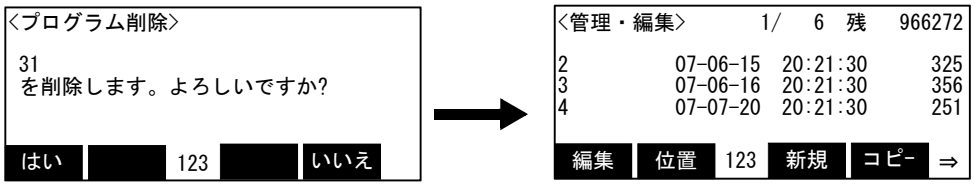


プログラム名称の設定 [3] [1] [EXE]

この場合、プログラム名称 ″31″ を削除します。

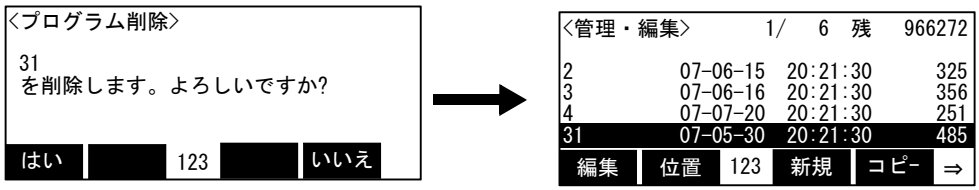
3) プログラムの削除

″ はい ″ に対応するファンクションキーを押すと指定プログラムを削除し、プログラム一覧表示に戻ります。
削除しない場合は ″ いいえ ″ に対応するファンクションキーを押してください。削除画面に戻ります。



削除する [F1]

この場合、プログラム名称 ″31″ を削除します。



削除しない [F4]

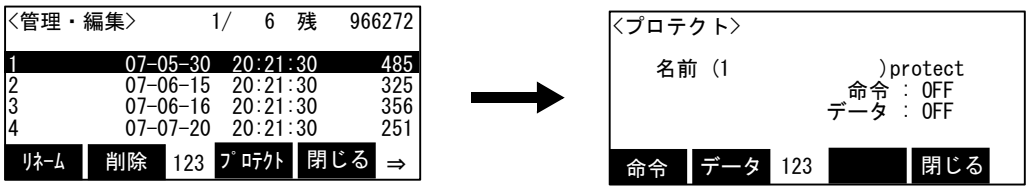
この場合、プログラムは削除されません。

◇◆◇保護 (プロテクト) されているプログラムは削除できません◇◆◇
命令プロテクトが設定されているプログラムは削除できません。命令プロテクトを解除してから実行してください。命令プロテクトの解除方法は「(5) プログラムの保護 (プロテクト)」を参照してください。

(5) プログラムの保護 (プロテクト)

1) プロテクトメニューの選択

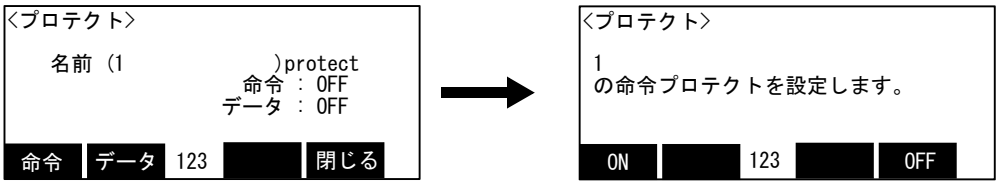
プログラム一覧表示で、″プロテクト″に対応するファンクションキー([F3]) を押します。プロテクト画面を表示します。″プロテクト″メニューが表示されていない場合は、[FUNCTION] キーを押して表示させます。



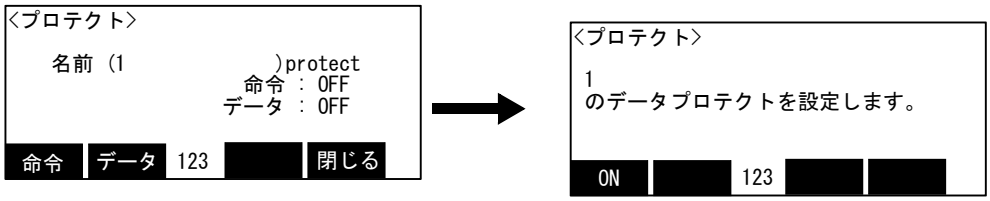
メニューの選択 [F3]

2) プロテクトの設定

プログラムの保護は、その命令文とデータ (変数値) を別々に指定することができます。
命令文の保護指定を設定する場合は″命令″に対応するファンクションキーを、データの保護指定を設定する場合は″データ″に対応するファンクションキーを押します。

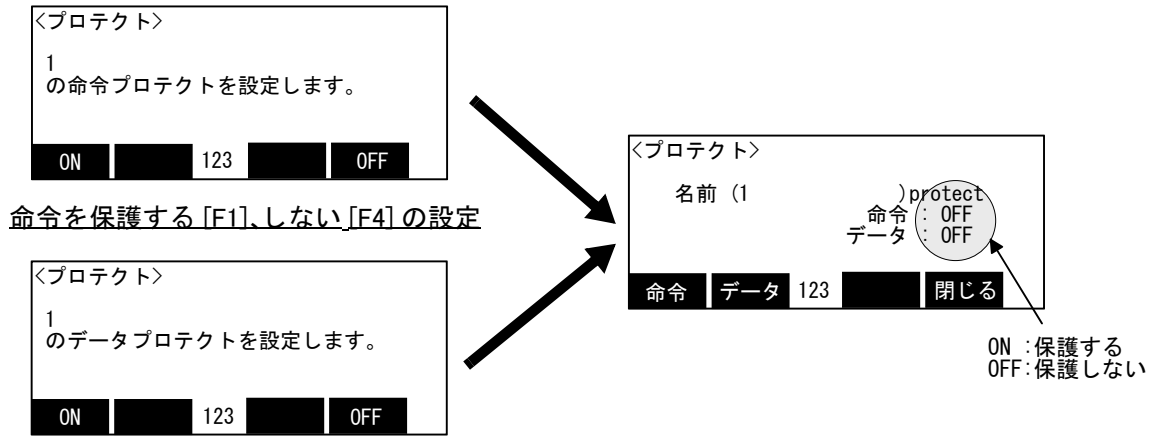


命令文の保護指定 [F1]



データの保護指定 [F2]

それぞれ″ON″に対応するファンクションキーを押すと″保護をする″、″OFF″に対応するファンクションキーを押すと″保護をしない″に設定されます。



◇◆◇命令プロテクトとは◇◆◇
プログラムの削除、名称変更、および命令の変更を誤操作から護する機能です。
・コピー操作では、保護情報はコピーされません。
・初期化操作では、保護情報を無視して初期化を実行します。

◇◆◇データプロテクトとは◇◆◇
誤操作による位置データの登録、変更や、誤ったプログラムの実行による各変数への代入から変数を保護する機能です。
・コピー操作では、保護情報はコピーされません。
・初期化操作では、保護情報を無視して初期化を実行します。

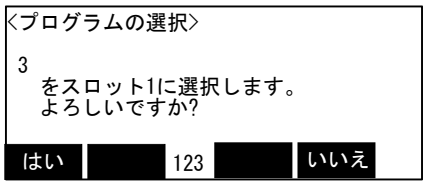
(6) プログラムの選択
本機能は T/B のソフトウェアバージョン 1.3 以降で可能です。
ステップ運転や自動運転の対象とするプログラムを選択することができます。
コントローラの操作パネルでの「3.7.2 プログラムの選択」と同じ機能で、プログラム選択後は操作パネルの "STATUS NUMBER" のプログラム番号にも表示されます。
以下に操作方法を示します。

1) プログラム選択
プログラム一覧表示で、[↑]、[↓] キーを操作して選択するプログラムにカーソルを合わせます。

プログラム選択 [↑],[↓]

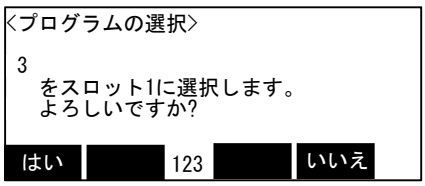
図はプログラム 3 を選択した例です。

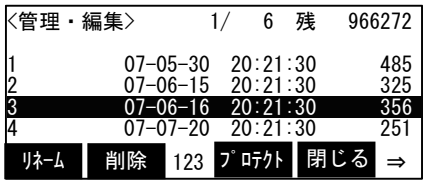
2) 選択プログラムの決定
T/B の [FUNCTION]+[EXE] キーを押します。確認画面が表示されます。



選択プログラムの決定 [FUNCTION]+[EXE]

表示されているプログラム名 (番号) を確認し，選択を実行する場合は " はい " に対応するファンクションキー ([F1]) を押します。
中止する場合は " いいえ " に対応するファンクションキー ([F4]) を押してください。いずれもプログラム一覧表示に戻ります。





プログラム選択の実行: [F1] (はい)
中止: [F4] (いいえ)

以上でプログラムの選択は終了です。

3.12 運転画面の操作

ここでは、以下の機能の操作方法を説明します。

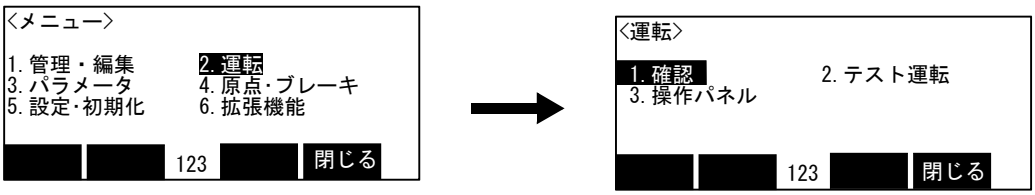
(1) 実行行の表示.....1. 確認　　: 実行中のプログラムの実行行の表示やステップ運転をおこないます。

(2) テスト運転画面.....2. テスト運転: 選択されているプログラムの名称、および実行中のステップ番号を表示します。また、プログラム実行中に連続運転モードをサイクル停止モードに切り替えます。

3.12.1 実行行の表示

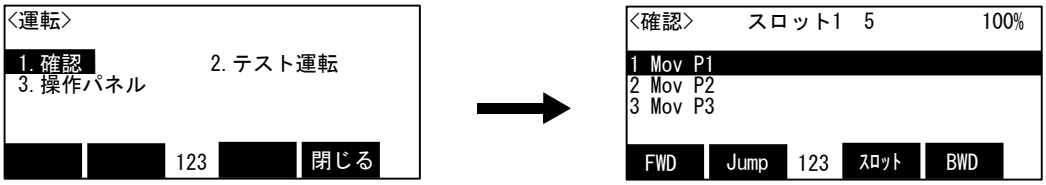
(1) 確認画面の選択

1) メニュー画面で [2] キーを押すと、運転メニュー画面を表示します。



運転メニュの選択 [2]

2) メニュー画面で [1] キーを押すと、確認画面を表示します。
画面にはスロット 1 に設定されているプログラムを表示します。スロット番号に続いてプログラム名称が表示されます。



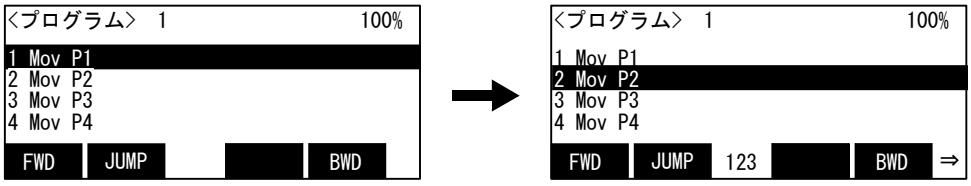
確認の選択 [1]

プログラム実行中は実行行にカーソルが移動します。

(2) ステップ運転

46 ページの「3.6 デバッグ操作」に示したステップ送りやステップ戻しと同様の操作がおこなえます。

1) ステップ送り
確認画面で "FWD" に対応するファンクションキー ([F1]) を押すと、押している間だけカーソルがあるステップを実行します。
ファンクションキーを途中で離すと、実行は中断します。また、ロボットの移動中はイネーブルスイッチ (3 ポジションスイッチ) を離すか、さらに強く押し込むことサーボ OFF し、実行を中断することができます。
実行中はオペレーティングパネルの [START] スイッチの LED が点灯します。1 ステップの実行を完了すると [START] スイッチの LED は消灯し、[STOP] スイッチの LED が点灯します。T/B 画面のカーソルは次のステップへ移動します。



ステップ送り [F1]

ファンクションキー ([F1]) を押す毎に次のステップへ進みます。

⚠ 注意　動作中はロボットの動きに十分注意してください。周辺装置との干渉など異常時にはファンクションキーを離すか、イネーブルスイッチ(3 ポジションスイッチ)を離すか強く押さえてロボットを停止させてください。

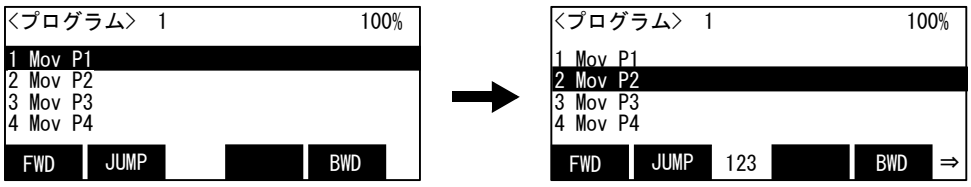
◇◆◇ステップ運転とは◇◆◇
プログラムを 1 ステップずつ実行することをステップ運転といいます。動作速度が遅く、1 ステップごとに実行が停止するので、プログラムの確認や動作位置の確認作業に有効です。
カーソルのあるステップが実行されます。
END 命令や HLT 命令を実行すると、それ以上ステップ送りは進みません。

◇◆◇実行ステップの変更◇◆◇
矢印キーでのカーソル移動や、ジャンプ操作(″JUMP″)で実行ステップを変更できます。

◇◆◇動作中のロボットを直ちに停止するには◇◆◇
① [EMG. STOP] スイッチを押す。
サーボ OFF し、非常停止エラーが発生して動作中のロボットは直ちに停止します。
継続は、エラーリセット → サーボ ON → ステップ運転操作の順におこないます。
②イネーブルスイッチ(3 ポジションスイッチ)を離す、またはさらに強く押し込む。
サーボ OFF し、エラー 2000 が発生して動作中のロボットは直ちに停止します。
継続は、エラーリセット → サーボ ON → ステップ運転操作の順におこないます。

2) ステップ戻し
ステップ送りや自動運転で実行の終わったステップ、あるいは中断しているステップを、補間命令に限り 1 ステップずつ戻し方向に実行します。なお、戻すことができるのは 4 ステップまでです。

″BWD″ に対応するファンクションキー([F4])を押すと、押し続けている間だけステップの戻り方向へ 1 ステップだけ実行します。
ファンクションキーを途中で離すと、実行は中断します。また、ロボットの移動中はイネーブルスイッチ(3 ポジションスイッチ)を離すか、さらに強く押し込むことサーボ OFF し、実行を中断することができます。
実行中はオペレーティングパネルの [START] スイッチの LED が点灯します。1 ステップの実行を完了すると [START] スイッチの LED は消灯し、[STOP] スイッチの LED が点灯します。T/B 画面のカーソルはステップの戻り方向で次の補間命令のステップへ移動します。



ステップ送り [F4]

ファンクションキー([F4])を押す毎に前のステップへ戻ります。

⚠ 注意　動作中はロボットの動きに十分注意してください。周辺装置との干渉など異常時にはファンクションキーを離すか、イネーブルスイッチ(3 ポジションスイッチ)を離すか強く押さえてロボットを停止させてください。

◇◆◇動作中のロボットを直ちに停止するには◇◆◇

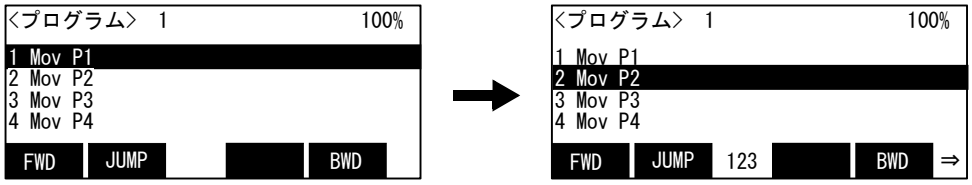
① [EMG. STOP] スイッチを押す。
サーボ OFF し、非常停止エラーが発生して動作中のロボットは直ちに停止します。
継続は、エラーリセット → サーボ ON → ステップ運転操作の順におこないます。

②イネーブルスイッチ (3 ポジションスイッチ) を離す、またはさらに強く押し込む。
サーボ OFF し、エラー 2000 が発生して動作中のロボットは直ちに停止します。
継続は、エラーリセット → サーボ ON → ステップ運転操作の順におこないます。

(3) ステップジャンプ

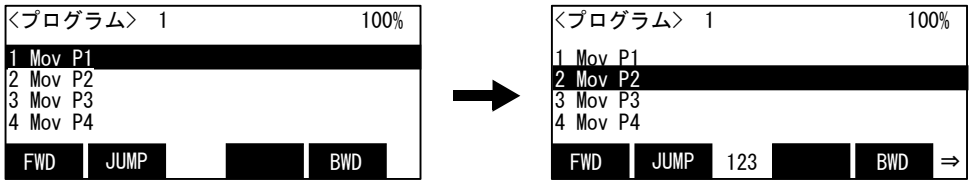
ステップ運転をおこなうステップ番号を変更することができます。
以下に例としてステップ 5 からステップ運転をおこなう場合の操作を示します。

1) ステップ 5 の呼び出し。
"JUMP" に対応するファンクションキー ([F2]) を押して表示されるステップ番号入力画面で、[5]、[EXE] キーを押します。ステップ 5 にカーソルが移動します。



ステップ送り [F2]

2) ステップ送り
前述 68 ページの「(2) ステップ運転」と同様に、"FWD" に対応するファンクションキー (F1]) を押すと、ステップ 5 からステップ送りをおこなうことができます。



ステップ送り [F1]

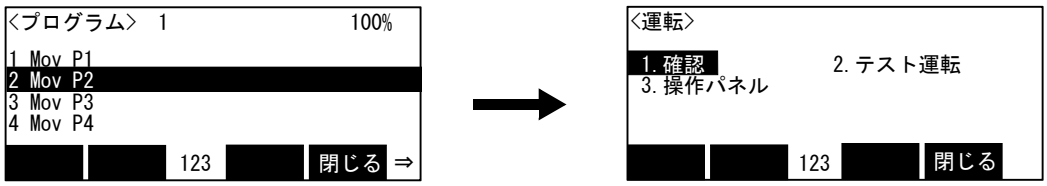
⚠ 注意 実行するステップを変更してステップ運転をおこなう場合は、ロボットが周辺装置などと干渉しないことを予め確認した上でおこなってください。
また、動作中はロボットの動きに十分注意し、周辺装置との干渉など異常時にはファンクションキーを離すか、イネーブルスイッチ (3 ポジションスイッチ) を離す、またはさらに強く押さえてロボットを停止させてください。

(4) 別スロットでのステップ送り

マルチタスクのプログラムを確認する場合は運転メニューの確認画面でステップ送りができます。
操作方法は 71 ページの「(4) 別スロットでのステップ送り」を参照願います。

(5) 確認画面の終了

1)〝閉じる〟に対応するファンクションキー([F4])を押すと、運転メニュー画面に戻ります。

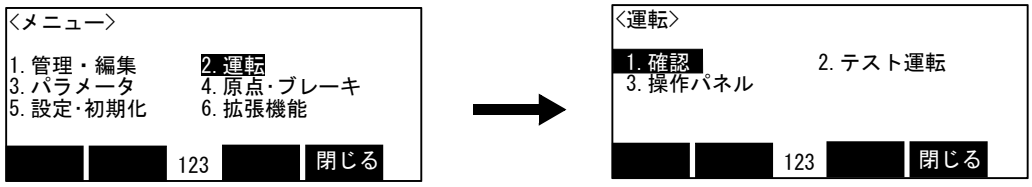


テスト運転画面の終了［F4］

3.12.2 テスト運転画面

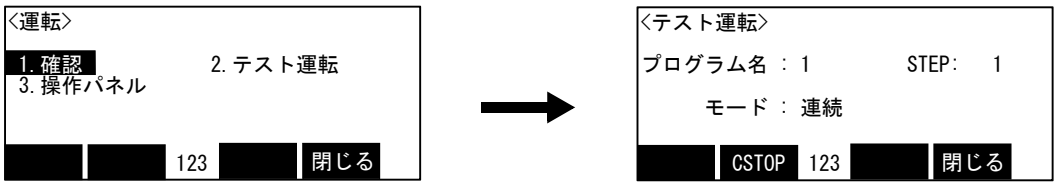
(1) テスト運転画面の選択

1) メニュー画面で [2] キーを押すと、運転メニュー画面を表示します。



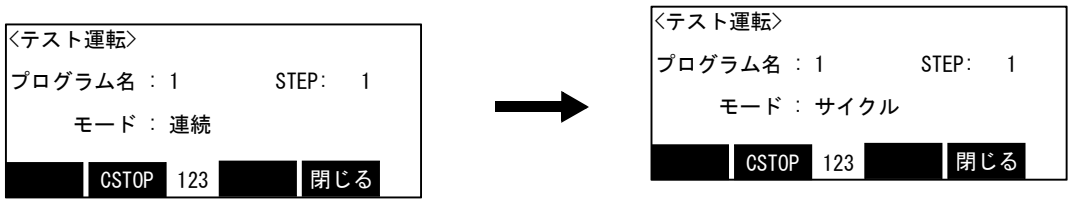
運転メニュの選択 [2]

2) 運転メニュー画面で [2] キーを押すと、テスト運転画面を表示します。
画面にはスロット 1 に設定されているプログラム名と実行ステップ番号、および現在の運転モードを表示します。



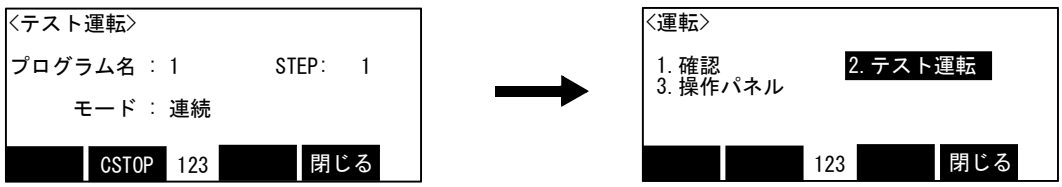
確認の選択 [2]

3) プログラム実行中に "CSTOP" に対応するファンクションキー ([F2]) を押すと、連続運転モードからサイクル運転モードへ切り替えることができます。モードの後に " サイクル " と表示するとともに、コントローラ操作パネルの [END] ボタンが点滅し、END 命令、またはプログラムの最終行を実行後にプログラムの運転を終了します。



サイクル運転ﾓｰﾄﾞへの変更 [F2]

4) " 閉じる " に対応するファンクションキー ([F4]) を押すと、運転メニュー画面に戻ります。



ﾃｽﾄ運転画面の終了 [F4]

◇◆◇プログラムの実行を停止すると連続運転モードになります◇◆◇
サイクル運転モード中に [STOP] キーを押すなどしてプログラムの実行を停止させた場合は、連続運転モードに切り替わります。サイクル運転モードでプログラムの実行を継続する場合は、[START] ボタンを押した後再度 [F4] キーを押してください。(コントローラ操作パネルの [END] ボタンを押すことでもサイクル運転モードになります)

3.12.3 操作パネル画面の操作

サーボ電源のオン / オフ、プログラムの選択や自動運転の起動などをおこないます。

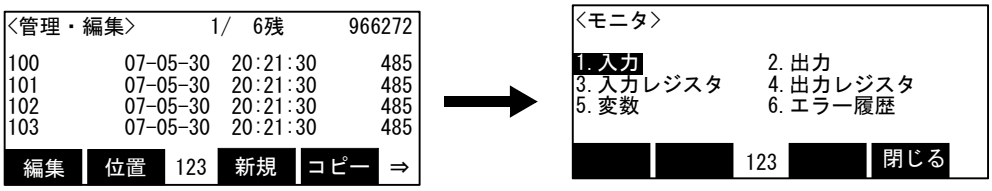
詳細を 53 ページの「3.7.3 自動運転の開始」中の「(2)T/B からの起動」に示しますので参照願います。

3.13 モニタ画面の操作

ここでは、以下の機能の操作方法を説明します。

(1) 入力信号のモニタ .....1. 入力　　: パラレル入力信号のモニタ。
(2) 出力信号のモニタ .....2. 出力　　: パラレル出力信号のモニタと ON/OFF の設定。
(3) 入力レジスタのモニタ .....3. 入力ﾚｼﾞｽﾀ : CC-Link 使用時、入力レジスタの値をモニタ。
(4) 出力レジスタのモニタ .....4. 出力ﾚｼﾞｽﾀ : CC-Link 使用時、出力レジスタの値をモニタ。
(5) 変数のモニタ .....5. 変数　　: プログラムで使用している変数値のモニタと値の設定。
(6) エラーの履歴表示 .....6. エラー履歴: 発生したエラーの履歴を表示。

上記いずれの操作も、T/B の [MONITOR] キーを押してモニタメニュー画面を表示させてからおこないます。T/B に操作権がなくても機能します。
その時表示されている画面は任意でかまいませんが、プログラム (命令) 編集画面では変数モニタは機能しません。



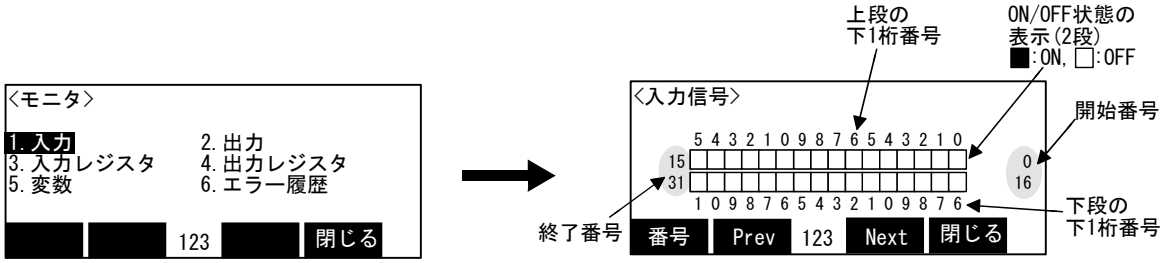
モニタ画面の選択 [MONITOR]

※ 表示されている画面は任意です。

(1) 入力信号モニタ

外部からの入力信号の ON/OFF 状態を常時モニタする機能です。

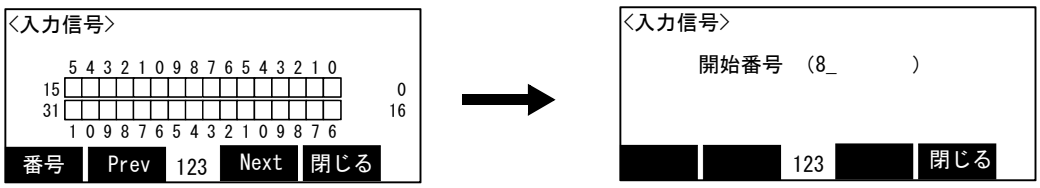
1) モニタメニュー画面で [1] キーを押すと、入力信号のモニタ画面を表示します。1 画面で 32 点の入力信号をモニタできます。



入力画面の選択 [1]

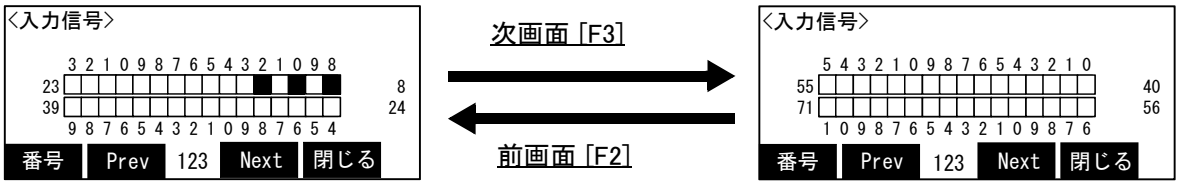
以下に入力信号の番号 8 から 15 の状態を確認する場合を示します。

2) ″ 番号 ″ に対応するファンクションキー [F1] を押します。
開始番号指定画面を表示しますので、開始番号に ″8″ を設定します。([8]、[EXE] キーを押します)



信号番号の設定 [F1]　　開始番号の設定 [8]、[EXE]

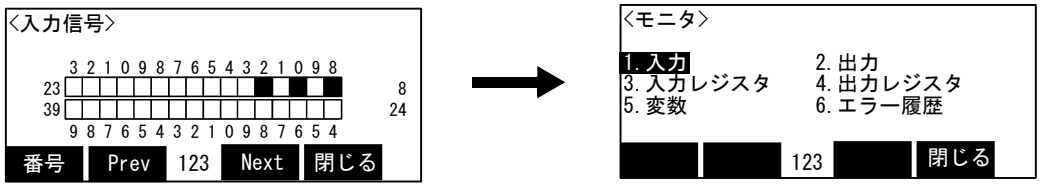
3) 入力信号 8 番を先頭に、32 点の ON/OFF 状態を表示します。黒塗りが ON、白抜きが OFF を表します。



図では、8,10,12 番が ON、他は OFF していることを表します。

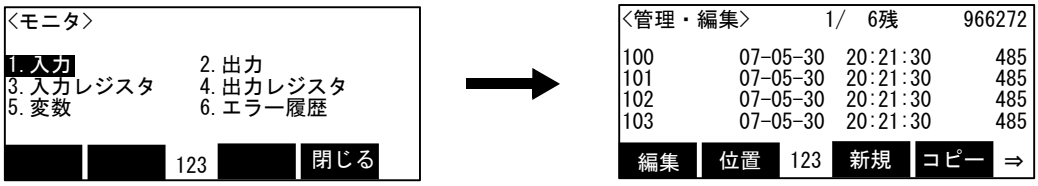
"Next" に対応するファンクションキー ([F3]) を押すと次の入力信号の状態 (次画面) を、"Prev" に対応するファンクションキー ([F2]) を押すと前の入力信号の状態 (前画面) を表示します。

4) " 閉じる " に対応するファンクションキー [F4] を押すと、モニタメニュー画面に戻ります。



信号番号の設定 [F4]

5) モニタメニュー画面で " 閉じる " に対応するファンクションキー [F4] を押すと、モニタを終了し元の画面に戻ります。



信号番号の設定 [F4]

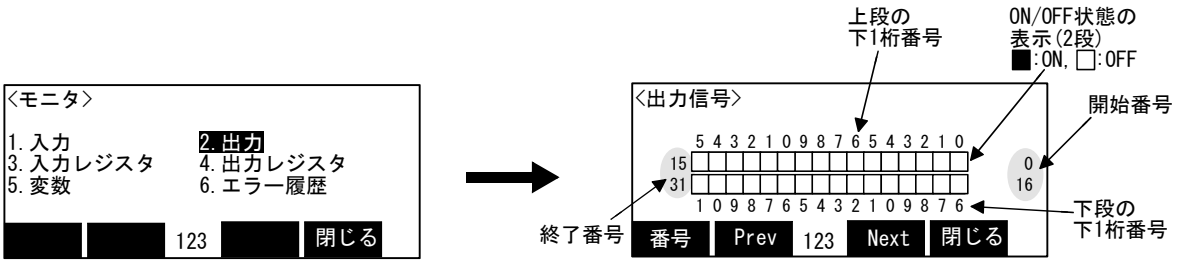
◇◆◇モニタの終了◇◆◇

入力信号の状態表示などのモニタ画面表示中に [MONITOR] キーを押すと、いつでもモニタを終了し元の画面に戻ります。

(2) 出力信号モニタ

外部への出力信号の ON/OFF 状態を常時モニタとする機能と、信号を強制的に出力する機能があります。

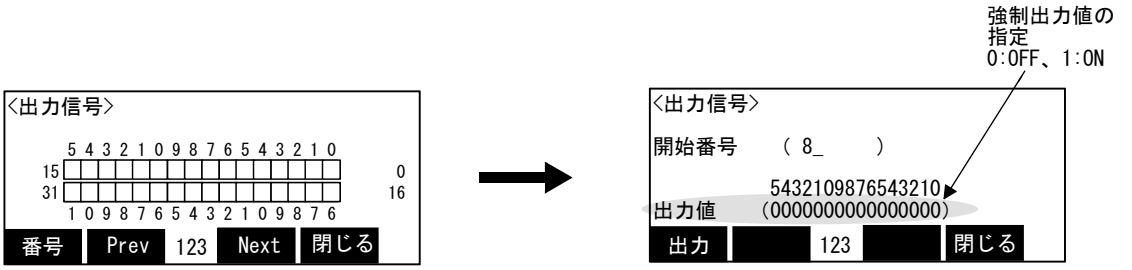
1) モニタメニュー画面で [2] キーを押すと、出力信号のモニタ画面を表示します。1 画面で 32 点の出力信号をモニタできます。



出力の選択 [2]

以下に出力信号の番号 8 から 15 の状態を確認する場合を示します。

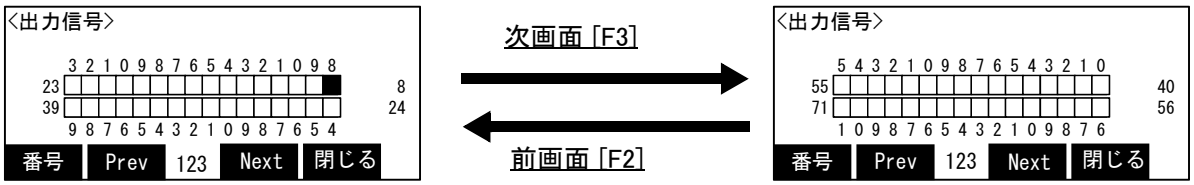
2) ″ 番号 ″ に対応するファンクションキー [F1] を押します。
開始番号と出力値を設定する画面を表示しますので、開始番号に ″8″ を設定します。([8]、[EXE] キーを押します)



信号番号の設定 [F1]

開始番号の設定 [8]、[EXE]

表示上の出力値には現在の出力信号の状態を表示していますが、ここは信号の強制出力値を設定する部分で常時表示にはなっていません。
″ 閉じる ″ に対応するファンクションキー ([F4]) を押すと、出力信号 8 番を先頭とした 32 点の出力信号モニタ画面を表示します。(常時表示)。黒塗りが ON、白抜きが OFF を表します。

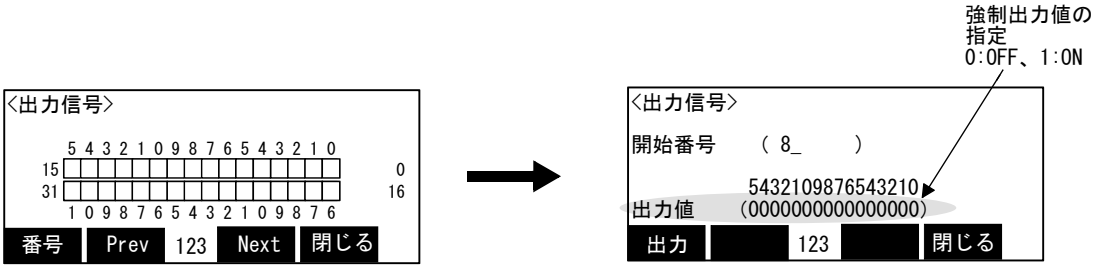


図では 8 番が ON、それ以外は
OFF していることを表します。

″Next″ に対応するファンクションキー ([F3]) を押すと次の入力信号の状態 (次画面) を、″Prev″ に対応するファンクションキー ([F2]) を押すと前の入力信号の状態 (前画面) を表示します。

3) 出力信号の強制出力

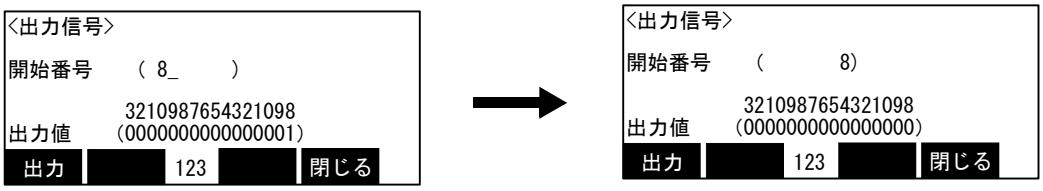
以下に、出力信号 8 番を強制的に OFF する場合の操作方法を示します。
″番号″に対応するファンクションキー[F1]を押します。
開始番号と出力値を設定する画面を表示しますので、開始番号に″8″を設定します。([8]、[EXE] キーを押します)



信号番号の設定 [F1]　　開始番号の設定 [8]、[EXE]

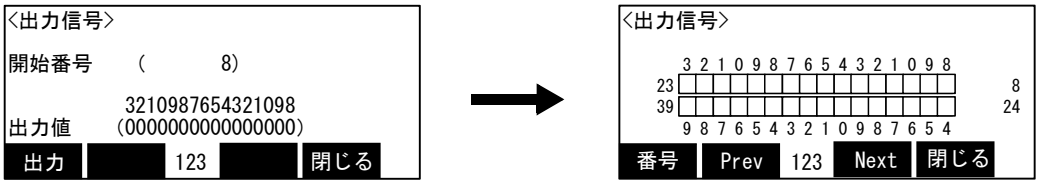
4) 矢印キーでカーソルを出力値の″8″の位置へ移動させます。

現在、出力信号 8 番号は ON しているので、値″1″を表示しています。
その値を OFF を示す″0″に変更して([0] キー)″出力″に対応するファンクションキー([F1])を押すと実際に同出力信号が OFF になります。



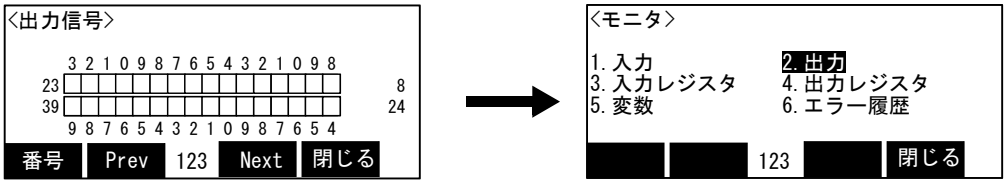
出力値の設定 [矢印]、[0]
設定値の出力 [F1]

5)″閉じる″に対応するファンクションキー([F4])を押すと、元の出力信号モニタ画面に戻ります。



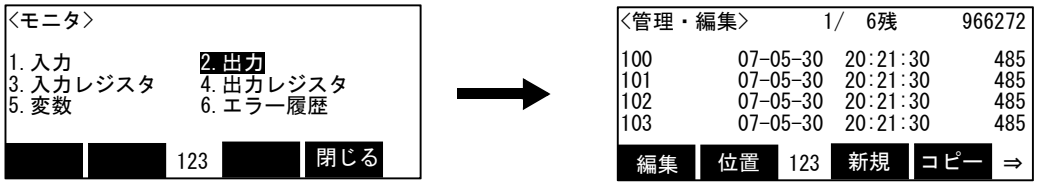
強制出力の終了 [F4]

6)″閉じる″に対応するファンクションキー[F4]を押すと、モニタメニュー画面に戻ります。



信号番号の設定 [F4]

7) モニタメニュー画面で " 閉じる " に対応するファンクションキー [F4] を押すと、モニタを終了し元の画面に戻ります。

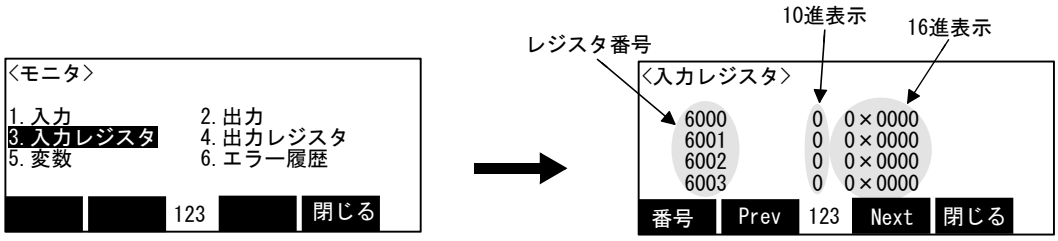


信号番号の設定 [F4]

(3) 入力レジスタモニタ

CC-Link 使用時に、その入力レジスタの値を常時モニタする機能です。
注)CR7xx-Q シリーズには CC-Link オプションがありませんので、本機能はご使用になれません。

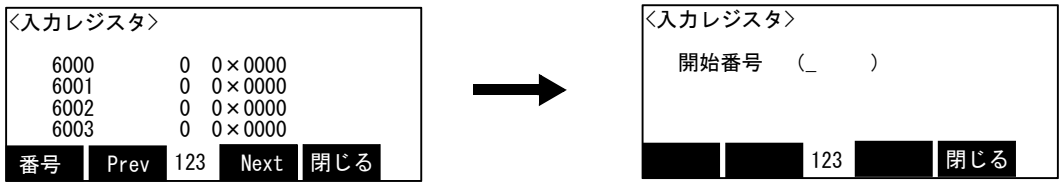
1) モニタメニュー画面で [3] キーを押すと、入力レジスタのモニタ画面を表示します。1 画面で 4 つの入力レジスタをモニタできます。



入力レジスタの選択 [3]

以下に入力レジスタ 8000 番の状態を確認する場合を示します。

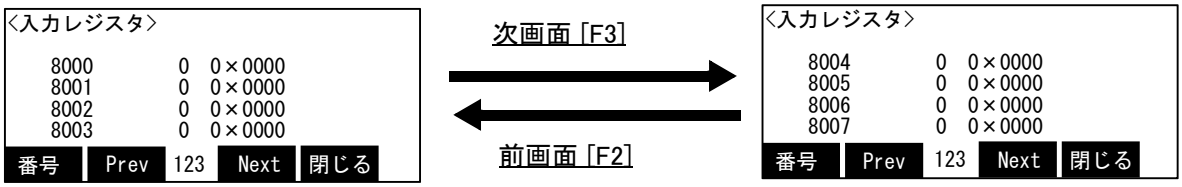
2) " 番号 " に対応するファンクションキー [F1] を押します。
開始番号指定画面を表示しますので、開始番号に "8000" を設定します。([8]、[0]、[0]、[0]、[EXE] キーを押します)



レジスタ番号の設定 [F1]

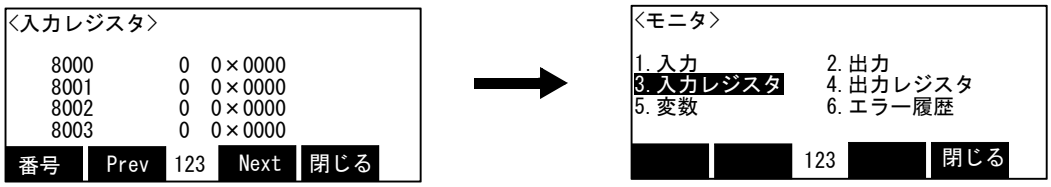
開始番号の設定 [8]、[0]、[0]、[0]、[EXE]

3) 入力レジスタ 8000 番を先頭に、4 つのレジスタの値を表示します。



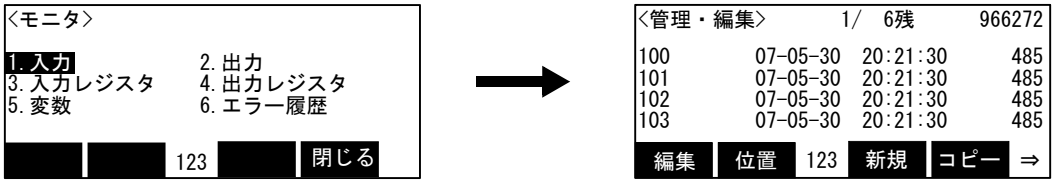
"Next" に対応するファンクションキー ([F3]) を押すと次の入力レジスタの状態 (次画面) を、"Prev" に対応するファンクションキー ([F2]) を押すと前の入力レジスタのの状態 (前画面) を表示します。

4)″閉じる″に対応するファンクションキー [F4] を押すと、モニタメニュー画面に戻ります。



終了 [F4]

5) モニタメニュー画面で″閉じる″に対応するファンクションキー [F4] を押すと、モニタを終了し元の画面に戻ります。



信号番号の設定 [F4]

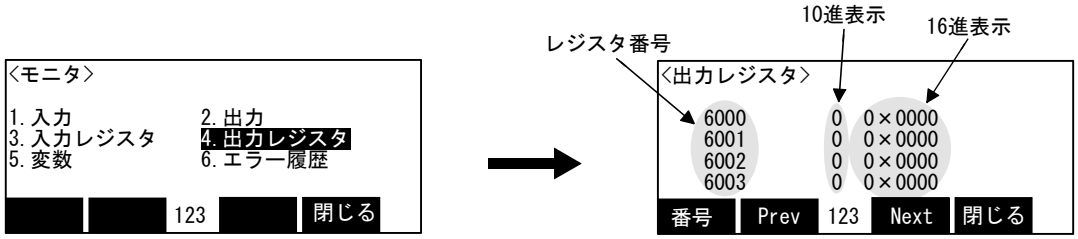
◇◆◇モニタの終了◇◆◇

入力レジスタの状態表示などのモニタ画面表示中に [MONITOR] キーを押すと、いつでもモニタを終了し元の画面に戻ります。

(4) 出力レジスタモニタ

CC-Link 使用時に、その出力レジスタの値を常時モニタする機能です。
注)CR7xx-Q シリーズには CC-Link オプションがありませんので、本機能はご使用になれません。

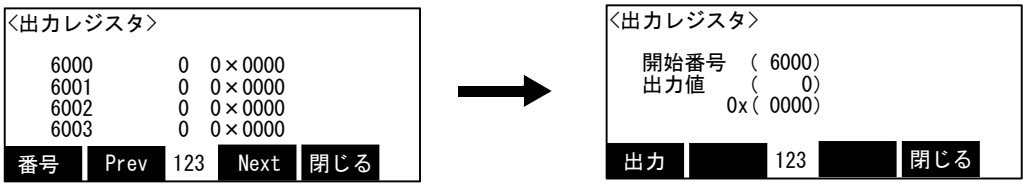
1) モニタメニュー画面で [4] キーを押すと、出力レジスタのモニタ画面を表示します。1 画面で 4 つの出力レジスタをモニタできます。



入力レジスタの選択 [4

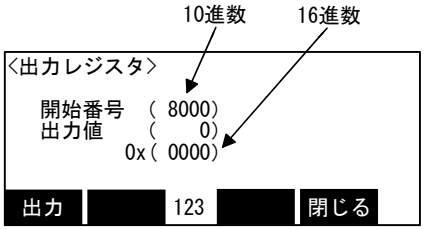
以下に出力レジスタ 8000 番の状態を確認する場合を示します。

2)″番号″に対応するファンクションキー[F1]を押します。
開始番号指定画面を表示しますので、開始番号に″8000″を設定します。([8]、[0]、[0]、[0]、[EXE]キーを押します)



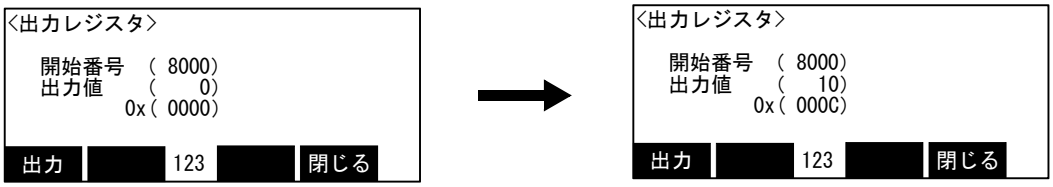
レジスタ番号の設定 [F1]　　開始番号の設定 [8]、[0]、[0]、[0]、[EXE]

3) 8000 番の現在の出力値が出力値に続くカッコ内に 10 進数で表示されます。下の 0x に続くカッコ内の値は 16 進数です。



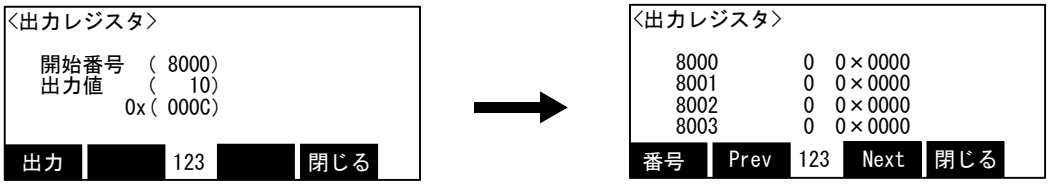
″閉じる″に対応するファンクションキー([F4])を押すと、出力レジスタの 8000 番を起点としたモニタ画面に戻すこともできますが、今の画面で出力値を変更することができます。
以下に出力レジスタ 8000 番の値を 12(10 進数)に設定する場合を示します。

4) 値の設定は 10 進数、または 16 進数で設定することができます。
10 進数で設定する場合は、矢印キーでカーソルを出力値へ移動させ、″10″を入力します。([1]、[0]キーを押します)不要な文字は[CLEAR]キーを押して消してください。
16 進数で設定する場合は矢印キーでカーソルを 0x へ移動させ、″C″を入力します。([C]キーを押します)不要な文字は[CLEAR]キーを押して消してください。
″出力″に対応するファンクションキー([F1])を押すと、設定された値を実際に出力します。



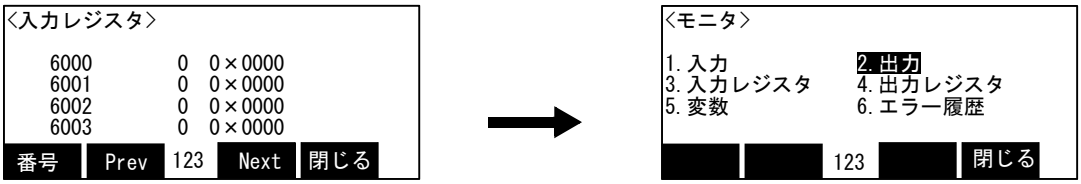
出力値の設定
10 進数で指定 [矢印]、[1]、[0]
16 進数で指定 [矢印]、[C]
設定値の出力 [F1]

5)″閉じる″に対応するファンクションキー([F4])を押すと、元の出力レジスタモニタ画面に戻ります。



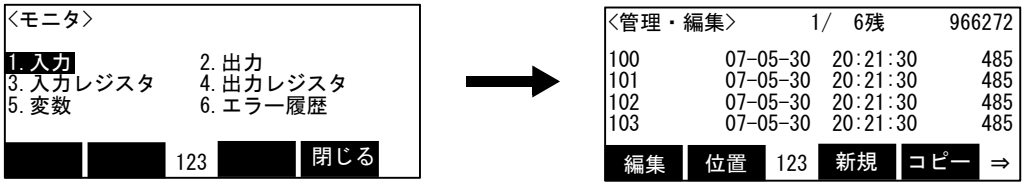
強制出力の終了 [F4]

6) もう一度 " 閉じる " に対応するファンクションキー ([F4]) を押すと、元のメニュー画面に戻ります。



出力レジスタモニタの終了 [F4]

7) モニタメニュー画面で " 閉じる " に対応するファンクションキー [F4] を押すと、モニタを終了し元の画面に戻ります。



信号番号の設定 [F4]

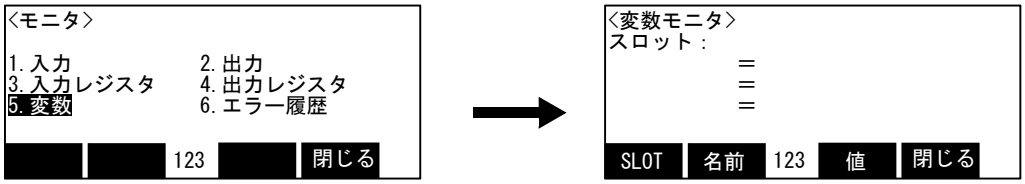
◇◆◇モニタの終了◇◆◇

出力レジスタの状態表示などのモニタ画面表示中に [MONITOR] キーを押すと、いつでもモニタを終了し元の画面に戻ります。

(5) 変数モニタ

プログラムで使用している変数の内容を表示、または変更する機能です。

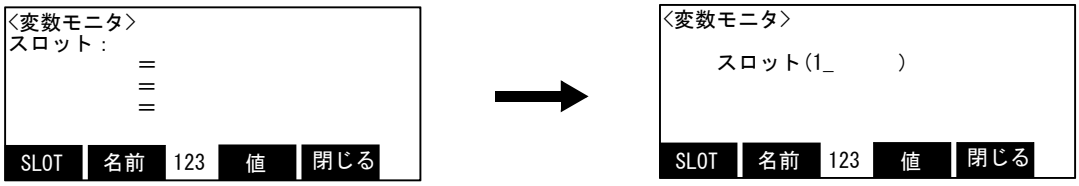
1) モニタメニュー画面で [5] キーを押すと、変数モニタ画面を表示します。



変数の選択［5］

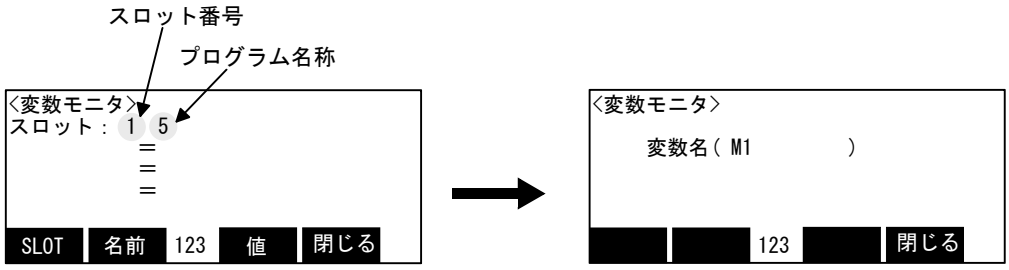
2) モニタの対象とするプログラムをスロット番号で指定します。
″SLOT″ に対応するファンクションキー ([F1]) を押して、スロット番号を入力します。

マルチタスク機能をご使用でない場合は ″1″ を設定してください。([1]、[EXE] キーを押します)



スロットの選択［F1］　　スロット番号の設定［1]、[EXE]

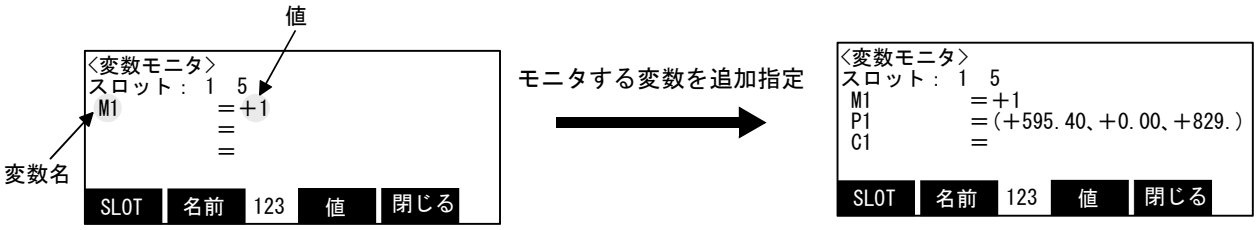
3) ″ スロット ：″ の後に、スロット番号とプログラム名称を表示します。
″ 名前 ″ に対応するファンクションキー ([F2]) を押して、モニタする変数名を入力します。



名前の選択［F2］

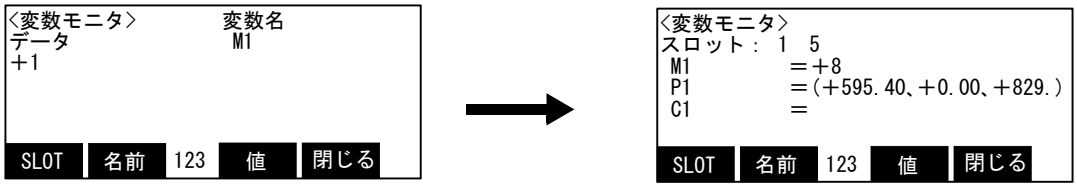
変数名の入力［M]、[1]、[EXE]
(数値変数 M1 をモニタする場合)

4) 画面に数値変数 M1 の値を表示します。
矢印キーを押して空いている行へカーソルを移動させ、上記 3) の操作を繰り返すとモニタする変数を追加できます。同時にモニタできる変数は 3 個までです。



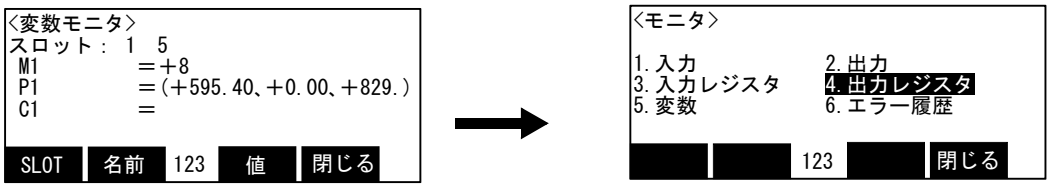
名前の選択［F2］　　同時に 3 個までモニタが可能

5) 変数値の変更
表示されている変数の値を変更することができます。
矢印キーを押して変更する変数名にカーソルを移動させ、″値″に対応するファンクションキー([F3])を押します。
現在の値(データ)を表示しますが、入力して変更することができます。



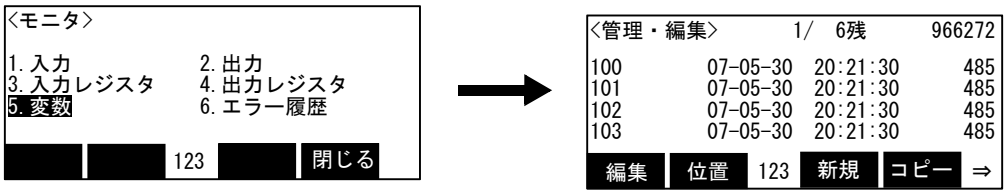
値の設定 [8]、[EXE]
(値を 8 に設定する場合)

6)″閉じる″に対応するファンクションキー([F4])を押すと、元のメニュー画面に戻ります。



変数タモニタの終了 [F4]

7) モニタメニュー画面で″閉じる″に対応するファンクションキー [F4] を押すと、モニタを終了し元の画面に戻ります。



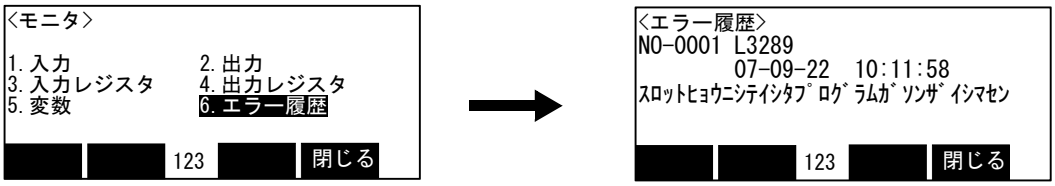
信号番号の設定 [F4]

◇◆◇操作権を問いません◇◆◇
本操作は T/B に操作権がなくてもおこなえます。
また、自動運転中でも変数の値(データ)を変更することができます。

(6) エラー履歴

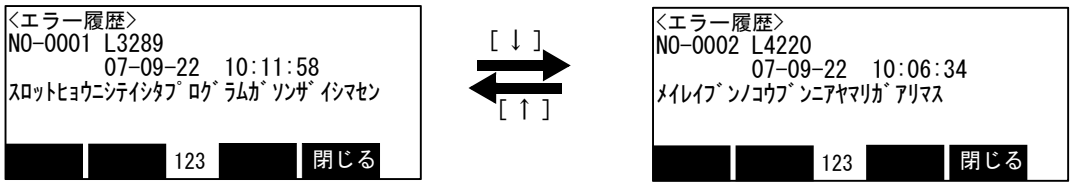
ロボットで発生したエラーの履歴を表示する機能です。トラブル発生時の参考にしてください。

1) モニタメニュー画面で [6] キーを押すと、エラー履歴画面を表示します。



変数の選択 [6]

矢印キーでエラー履歴の前後を表示します。



次のエラー履歴表示 [↓]、前のエラー履歴表示 [↑]

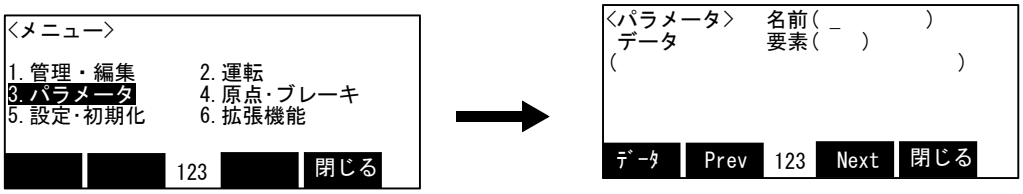
◇◆◇操作権を問いません◇◆◇

本操作は、T/B に操作権がなくてもおこなえます。

3.14 パラメータ画面の操作

パラレル I/O の専用入出力設定やツール長などの設定はパラメータとして登録されており、ロボットは各パラメータの設定値に基づいて動作します。その各パラメータの設定値を表示、登録する機能です。

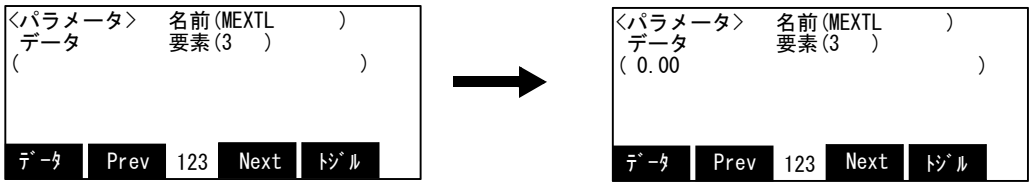
1) メニュー画面で [3] キーを押すと、パラメータ画面を表示します。



パラメータの選択 [3]

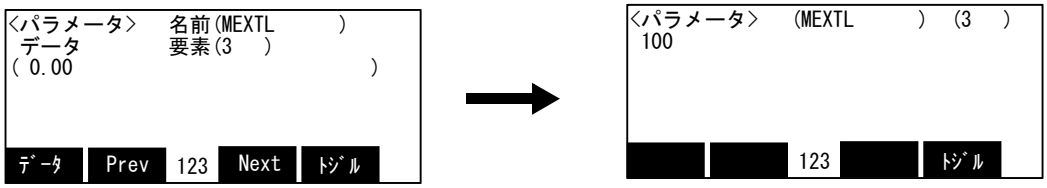
以下にパラメータ「MEXTL(ツールデータ)」の Z 軸 (3 番目の要素) の設定値を 0 から 100mm に変更する場合を示します。

2) 名前に "MEXTL" を、要素に "3" を入力します。
現在設定されているデータが表示されます。



名前の入力 [M]、[E]、[X]、[T]、[L]、[↓]
要素の入力 [3]
確定 [EXE]

3) " データ " に対応するファンクションキー ([F1]) を押して、新しい設定値 "100" を入力します。
不要な数字は [CLEAR] キーで削除してください。

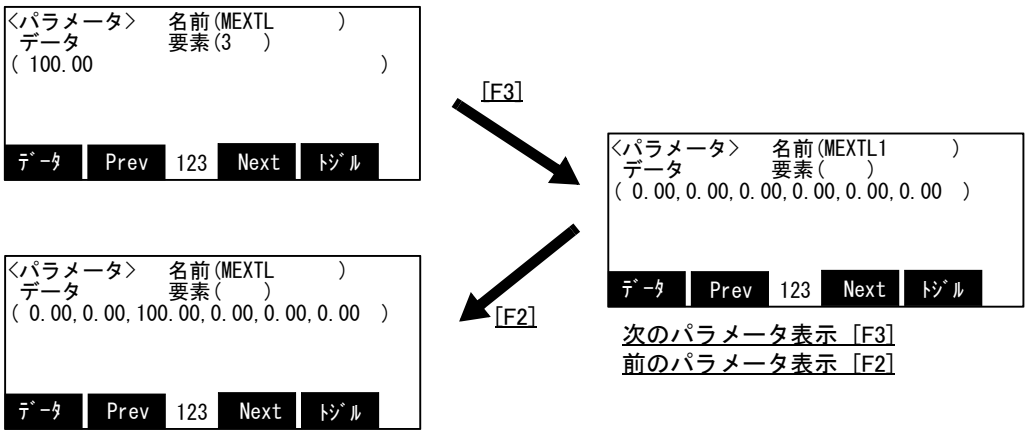


データの設定 [F1]

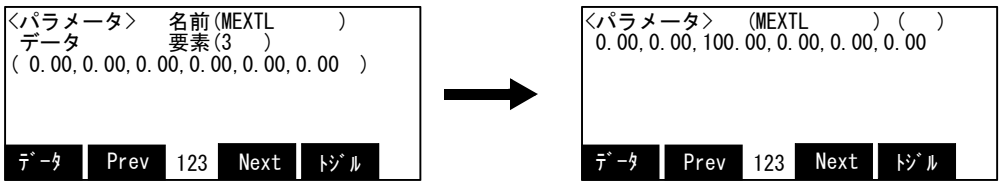
データの入力 [1]、[0]、[0]
確定 [EXE]
(キャンセルする場合は [F4])

[EXE] キーを押すとブザーが鳴って値を確定し、パラメータの表示画面に戻ります。
新しい設定値を入力した後でも、" トジル " に対応するファンクションキー ([F4]) を押すことで、設定値を変更せずにパラメータ表示画面に戻すことができます。

また、"Next" に対応するファンクションキー ([F3]) を押すと次のパラメータを、"Prev" に対応するファンクションキー ([F2]) を押すと前のパラメータを表示します。
この場合、名前で示されるパラメータのすべての要素を表示し、そのパラメータ単位で前後表示するため、要素番号の指定を削除します。



この状態でも値を変更することができます。
″データ″に対応するファンクションキー([F1])を押して、カーソルを変更する要素番号の位置へ矢印キーで移動させ、新しい設定値を入力します。不要な数字は[CLEAR]キーで削除してください。



データの設定 [F1]

カーソル移動［矢印］
データの入力 [1]、[0]、[0]
確定 [EXE]
(キャンセルする場合は[F4])

[EXE]キーを押すとブザーが鳴って値を確定し、パラメータの表示画面に戻ります。
新しい設定値を入力した後でも、″トジル″に対応するファンクションキー([F4])を押すことで、設定値を変更せずにパラメータ表示画面に戻すことができます。

◇◆◇電源の入れ直しが必要です◇◆◇
コントローラの電源を入れ直すことにより変更したパラメータが有効になります。

◇◆◇プログラム実行中は表示のみ可能◇◆◇
プログラムの実行中にパラメータの設定値を変更すると、エラーが発生します。(エラーが発生してもプログラムの実行は停止しません)

◇◆◇入力されたパラメータの名前に近いパラメータを表示◇◆◇
パラメータの名前はすべての文字を正確に入力しなくても、入力された名前に近いパラメータを自動で表示します。
これにより、長い名前のパラメータはその一部を入力しておいて、″Next″、または″Prev″に対応するファンクションキーを押して目的とするパラメータを表示させることができます。

3.15 原点・ブレーキ画面の操作

(1) 原点

パラメータの消失やロボットの干渉等により、原点位置が消失あるいはずれてしまった場合は、本操作によりロボットの原点を設定し直す必要があります。
操作方法は別冊の「取扱説明書 / ロボット本体セットアップから保守まで」を参照してください。

(2) ブレーキ

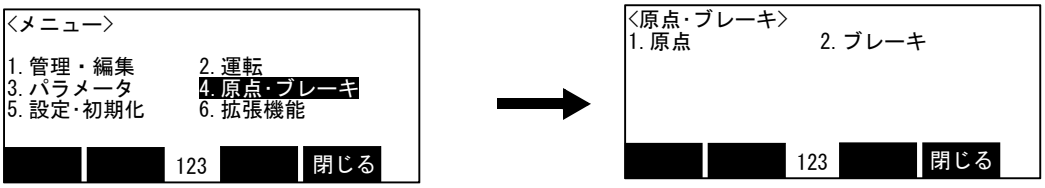
サーボオフしている状態で、サーボモータのブレーキを解除する機能です。サーボオフ操作は 60 ページの「3.8 サーボオン / オフ」を参照しておこなってください。
ロボットアームを手で直接動かす場合などに使用します。
本操作は、お客様でご用意のイネーブリングデバイスをコントローラの非常停止入力コネクタ (CNUSR) のイネーブリングデバイス入力端子に接続していないと有効になりません。
安全に操作するため、必ずイネーブリングデバイスを接続してください。

⚠注意

ブレーキ解除をおこなうと、ロボットの構成、解除する軸によりロボットアームが自重で落下します。
あらかじめ T/B の操作者とは別の人でアームが落下しないような処置をしておき、T/B の操作者がお互いの合図のもとに本操作をおこなうようにしてください。

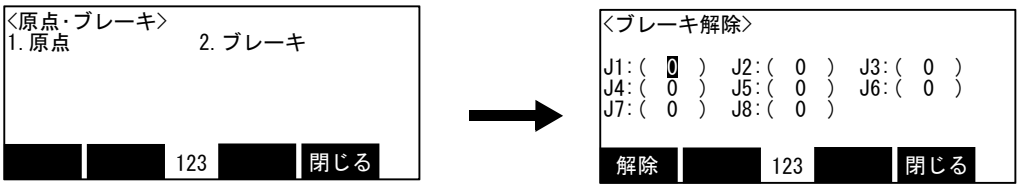
以下に操作方法を示します。本操作は、イネーブルスイッチ (3 ポジションスイッチ) を軽く押した状態で行ってください。

1) メニュー画面で [4] キーを押すと、原点・ブレーキ画面を表示します。



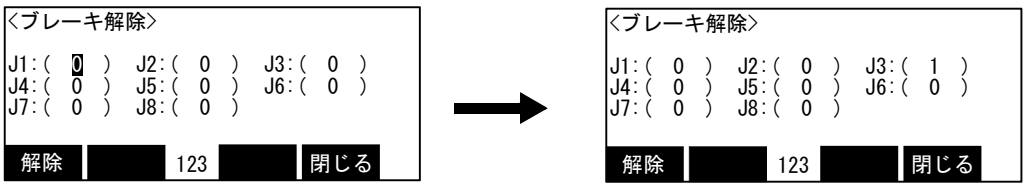
原点・ブレーキ画面の選択 [4]

2) 原点・ブレーキ画面で [2] キーを押すと、ブレーキ解除画面を表示します。



原点・ブレーキ画面の選択 [2]

3) ブレーキ解除する軸に "1" を入力します。

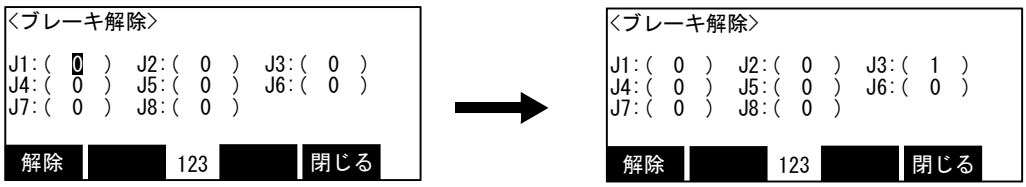


ブレーキ解除軸の設定 [矢印]、[1]

⚠注意　ブレーキ解除は複数の軸を同時に設定できますが、ロボットの軸構成によりその合成の自重でアームが落下します。
あらかじめ T/B の操作者とは別の人でアームが落下しないような処置をしておき、T/B の操作者がお互いの合図のもとに本操作をおこなうようにしてください。

4) イネーブルスイッチ(3 ポジションスイッチ)を軽く押した状態で、″解除″に対応するファンクションキー([F1])を押すと、ファンクションキー([F1])を押している間だけ指定した軸のブレーキが解除されます。
- RH-3FH シリーズ、RV-2F/4F/7F シリーズ、RH-3FHR シリーズ：
  ブレーキ解除は連続的です。
- RH-6FH/12FH/20FH シリーズ：
  ブレーキは断続的に解除 / ロックを繰り返します。
- RV-13F シリーズ、RV-20F シリーズ、RV-50F シリーズ：
  J2 軸、J3 軸ブレーキは断続的に解除 / ロックを繰り返します。他の軸のブレーキは連続的に解除します。

ブレーキを作動させるには、[F1] キーを離す、またはイネーブルスイッチ(3 ポジションスイッチ)をさらに強く押し込むか離してください。



ブレーキ解除［イネーブルスイッチ］、[F1]

3.16 設定・初期化画面の操作

ここでは、以下の機能の操作方法を説明します。

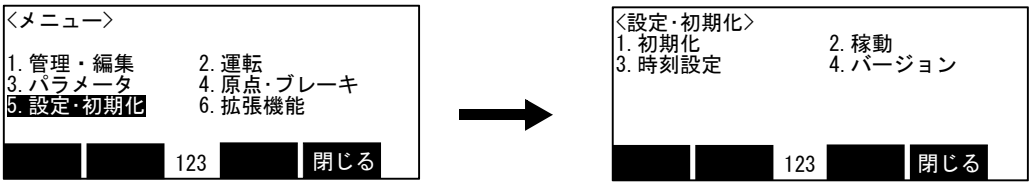
(1) 初期化 ..... 1. プログラム: すべてのプログラムを削除します。
2. パラメータ: パラメータを出荷時の設定に戻す。
3. バッテリ　: バッテリの消耗時間をリセットします。

(2) 稼動 ..... 電源 ON の累積時間と、バッテリの残り時間を表示します。

(3) 時刻設定 ..... 日付と時刻の表示、設定をおこないます。
ロボット CPU 内には時計機能があり、プログラム登録・変更時間やエラー時間の表示などに使用しています。現在の日付・時刻とずれているときは正しい日付、時間に変更してください。

(4) バージョン..... ロボット CPU とティーチングボックスのソフトウェアバージョンを表示します。

メニュー画面で [5] キーを押すと、設定・初期化メニュー画面を表示します。

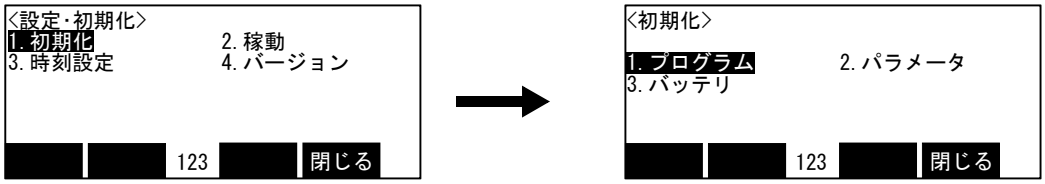


設定・初期化メニュー画面の選択［5］

(1) プログラムの初期化

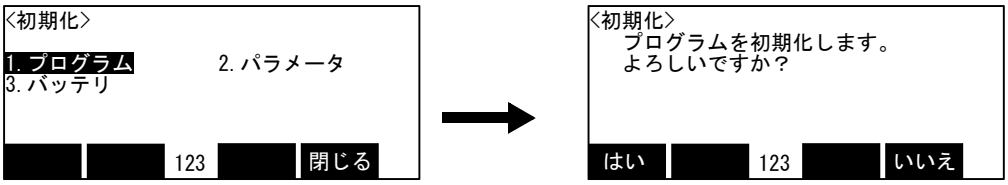
全てのプログラムを削除します。

1) 設定・初期化メニュー画面で [1] キーを押すと、初期化メニュー画面を表示します。



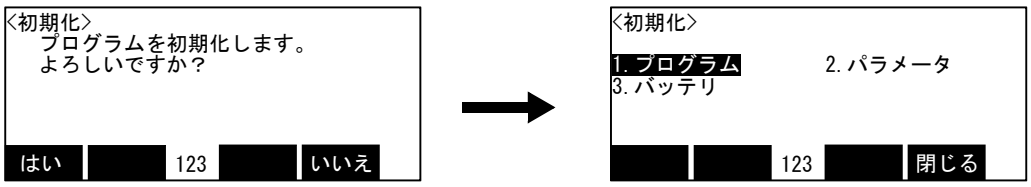
初期化メニュー画面の選択［1］

2) 初期化メニュー画面で [1] キーを押して〝プログラム〟を選択します。
確認の画面を表示します。



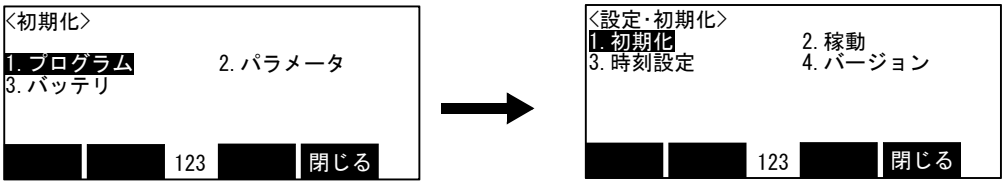
初期化メニュー画面の選択［1］

3) 初期化を実行する場合は、″はい″に対応するファンクションキー([F1])を、実行しない場合は″いいえ″に対応するファンクションキー([F4])を押します。
画面は初期化メニュー画面に戻ります。



初期化の確認
おこなう場合　　[F1]
おこなわない場合 [F4]

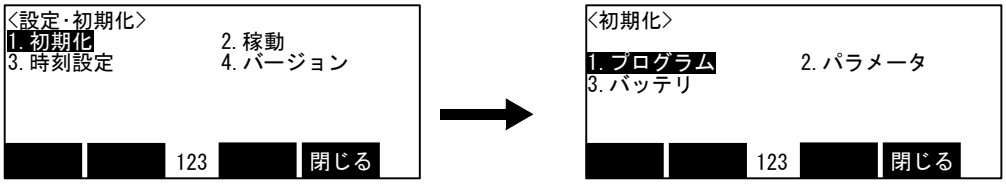
4) ″閉じる″に対応するファンクションキー([F4])キーを押すと、設定・初期化画面に戻ります。



初期化画面の終了 [F4]

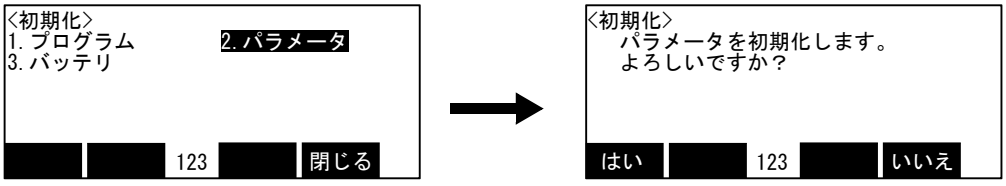
◇◆◇保護されていても実行します◇◆◇
プログラムの初期化は、プログラム保護、または変数保護が ON に設定されていても実行しますのでご注意ください。

(2) パラメータの初期化
パラメータを出荷時の設定に戻します。
1) 設定・初期化メニュー画面で [1] キーを押すと、初期化メニュー画面を表示します。



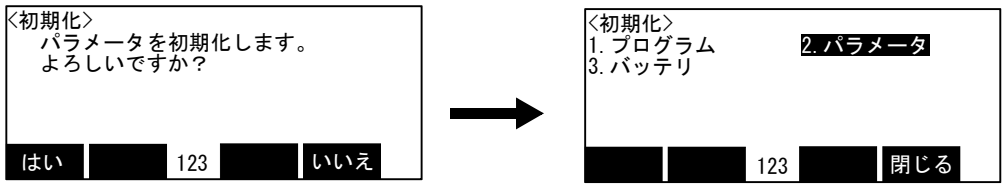
初期化メニュー画面の選択 [1]

2) 初期化メニュー画面で [2] キーを押して″パラメータ″を選択します。
確認の画面を表示します。



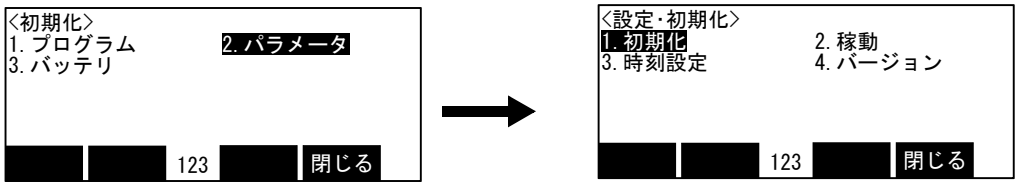
パラメータ選択 [2]

3) 初期化を実行する場合は、″はい″に対応するファンクションキー([F1])を、実行しない場合は″いいえ″に対応するファンクションキー([F4])を押します。
画面は初期化メニュー画面に戻ります。



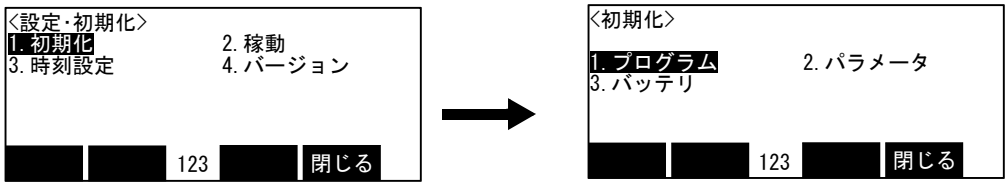
初期化の確認
おこなう場合　[F1]
おこなわない場合 [F4]

4)″閉じる″に対応するファンクションキー([F4])キーを押すと、設定・初期化画面に戻ります。



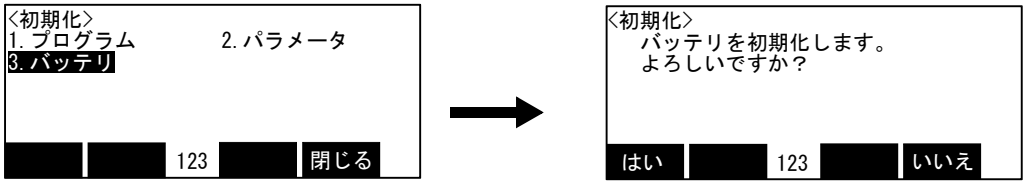
初期化画面の終了［F4］

(3) バッテリの初期化
バッテリの消耗時間をリセットします。
1) 設定・初期化メニュー画面で [1] キーを押すと、初期化メニュー画面を表示します。



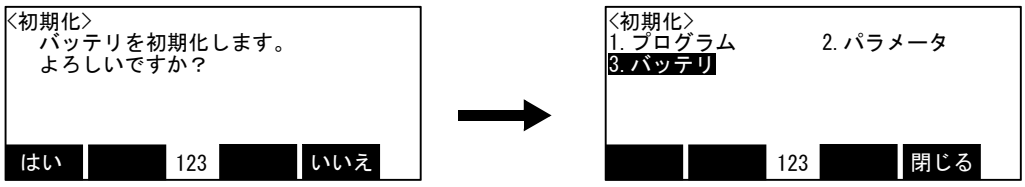
初期化メニュー画面の選択［1］

2) 初期化メニュー画面で [3] キーを押して″バッテリ″を選択します。
確認の画面を表示します。



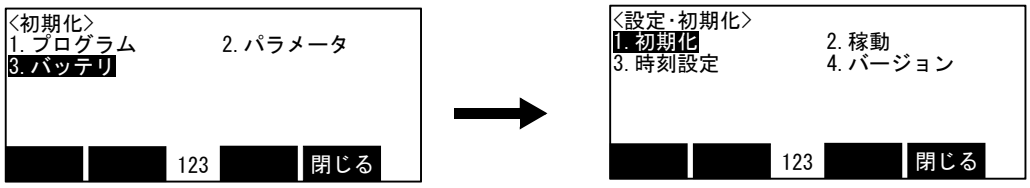
初期化メニュー画面の選択［1］

3) 初期化を実行する場合は、″はい″に対応するファンクションキー([F1])を、実行しない場合は″いいえ″に対応するファンクションキー([F4])を押します。
画面は初期化メニュー画面に戻ります。



初期化の確認
おこなう場合　[F1]
おこなわない場合 [F4]

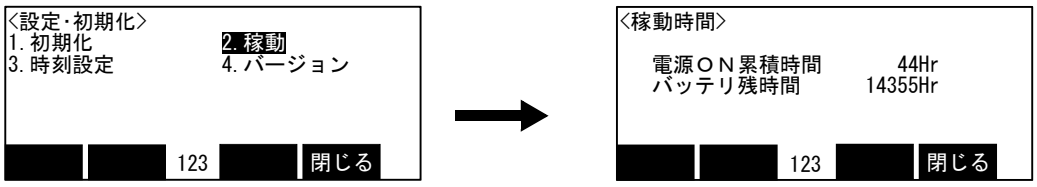
4) ″閉じる″に対応するファンクションキー([F4])キーを押すと、設定・初期化画面に戻ります。



初期化画面の終了 [F4]

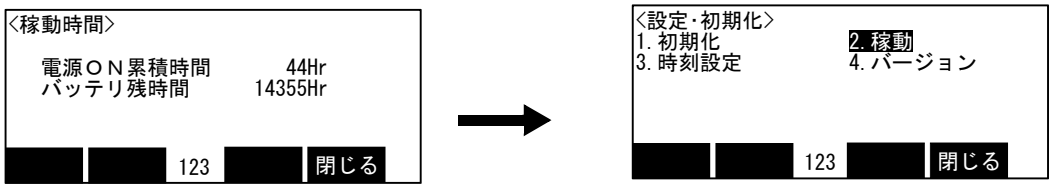
◇◆◇バッテリ交換時にのみ必ずおこなってください。◇◆◇
ロボット CPU 内部でバッテリの使用時間を計算しており、消耗時に注意メッセージを表示するようにしています。注意メッセージを正しく表示させるために、バッテリ交換時には必ず本初期化操作をおこなってバッテリ消耗時間を初期化してください。
また、バッテリを交換していない時に本初期化操作をおこなうと、注意メッセージの表示時期がずれてしまうので、バッテリ交換時のみおこなうようにしてください。

(4) 稼動
電源 ON の累積時間と、バッテリの残り時間を表示します。
1) 設定・初期化メニュー画面で [2] キーを押すと、稼動時間画面を表示します。



初期化メニュー画面の選択 [1]

2) ″閉じる″に対応するファンクションキー([F4])キーを押すと、設定・初期化画面に戻ります。

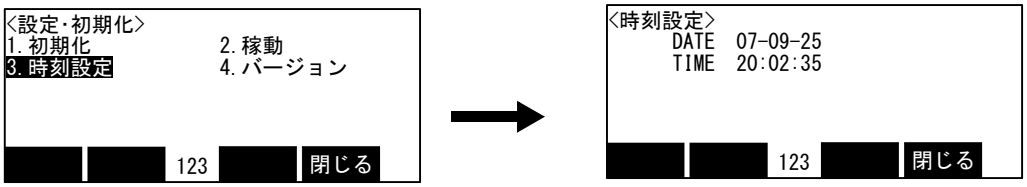


稼働時間画面の終了 [F4]

(5) 時刻設定

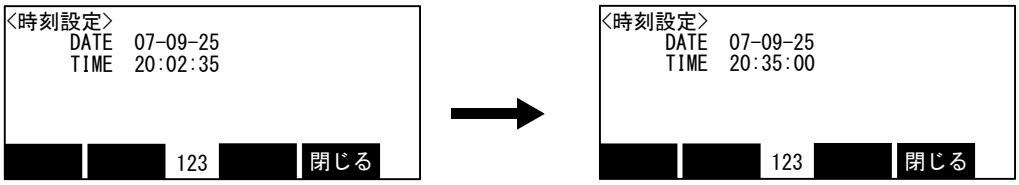
日付と時刻の表示、設定をおこないます。

1) 設定・初期化メニュー画面で [3] キーを押すと時刻設定画面が表示され、現在の日付と時刻を表示します。



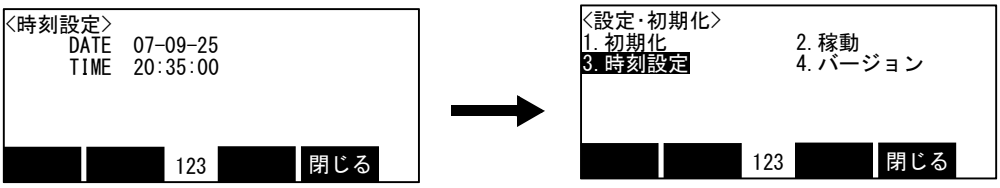
時刻設定画面の選択［3］

2) 時刻設定画面で日付と時刻を設定することができます。
矢印キーでカーソルを移動して、現在の日付と時刻を入力してください。



日付、時刻の入力［矢印］､［数値］
日付、時刻の確定［EXE］

3)〝閉じる〟に対応するファンクションキー([F4]) キーを押すと、設定・初期化画面に戻ります。

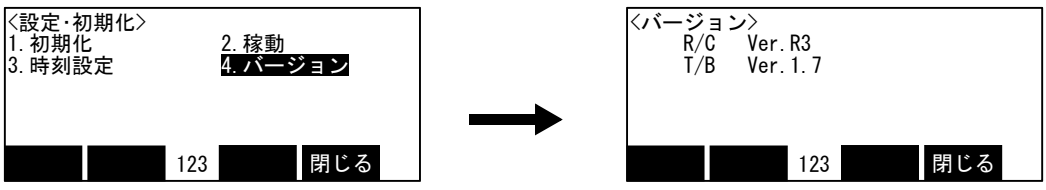


時刻設定画面の終了［F4］

(6) バージョン

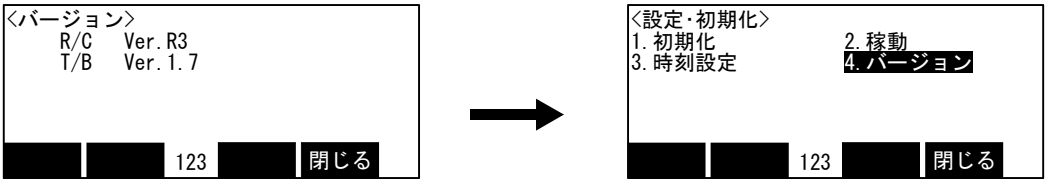
ロボット CPU とティーチングボックスのソフトウェアバージョンを表示します。

1) 設定・初期化メニュー画面で [4] キーを押すとバージョン画面を表示します。



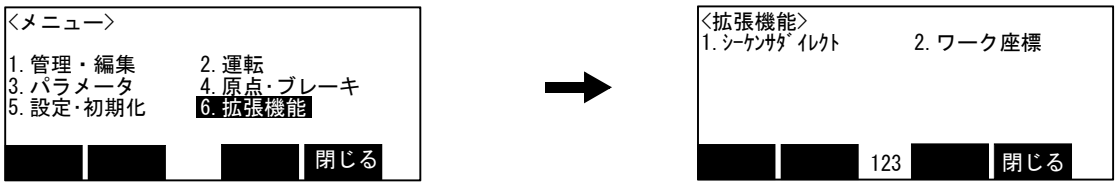
バージョン画面の選択［4］

2)〝閉じる〟に対応するファンクションキー([F4]) キーを押すと、設定・初期化画面に戻ります。



バージョン画面の終了［F4］

3.17 拡張機能の操作



メニューの選択 [6]

(1) シーケンサダイレクト

本機能は、シーケンサのプログラムでロボットを直接制御するものです。CR7xx-Q コントローラのみでご利用できます。
この仕様・操作方法の詳細は、別冊の「取扱説明書 /CR750-Q シリーズ、CRnQ-700 シリーズ iQ Platform 対応拡張機能説明書 (BFP-A8757)」を参照願います。」

(2) ワーク座標

ワークジョグ操作に必要なワーク座標系を定義する画面です。ワークジョグをご使用になる場合は、本画面で対象とするワーク座標系を定義してください。
操作方法の詳細は、ジョグ操作の説明箇所に記載しています。そちらを参照してご活用願います。
【参照箇所】

1) ワーク座標の設定、ワークジョグ操作：別冊の「取扱説明書 / ロボット本体セットアップから保守まで」
2) ジョグ送りの種類：本書 /22 ページの「3.2.1 ジョグ送りの種類」
3) 関連パラメータ　：本書 /408 ページの「5.1 動作パラメータ」の「ﾜｰｸ座標系」

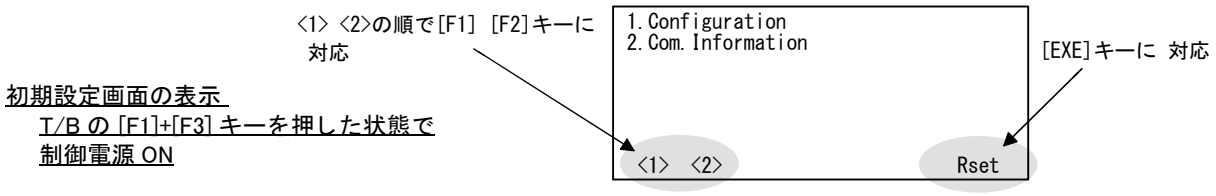
3.18 初期設定画面の操作

以下に示す初期設定の機能があります。

(1) 表示言語の設定.....T/B の画面に表示する文字を、日本語または英語のいずれかに設定できます。

(2) コントラスト調整.....T/B の画面の明るさを 16 段階で調整できます。

この操作は T/B の [F1] キーと [F3] キーを 2 重押しした状態で制御電源を ON して表示される初期設定画面で操作します。

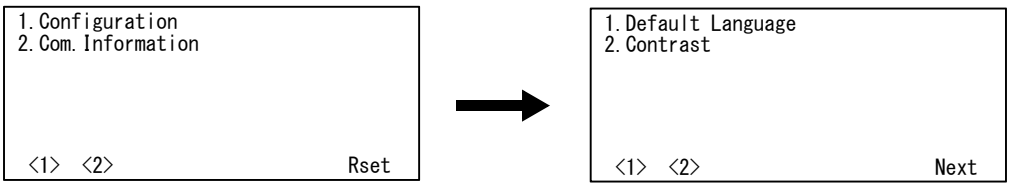


初期設定画面の表示
T/B の [F1]+[F3] キーを押した状態で制御電源 ON

(1) 表示言語の設定

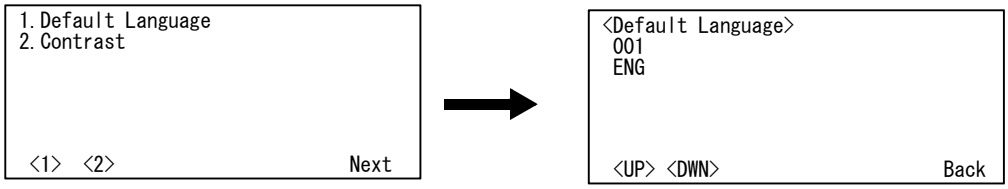
T/B の画面に表示する文字を、日本語または英語のいずれかに設定します。

1) 初期設定画面で [F1] キーを押して "1.Configuration" を選択します。



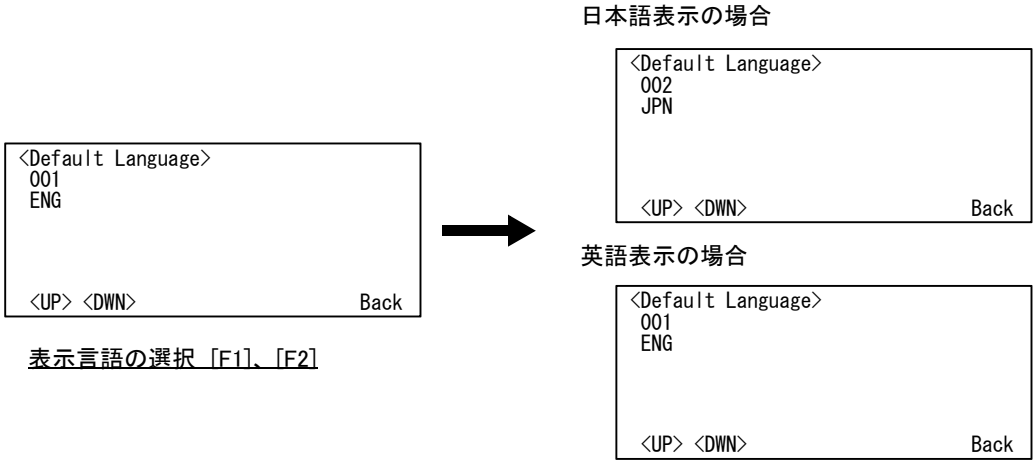
"1.Configuration" の選択 [F1]

2) [F1] キーを押して "1.Default Language" を選択します。



"1.Default Language" の選択 [F1]

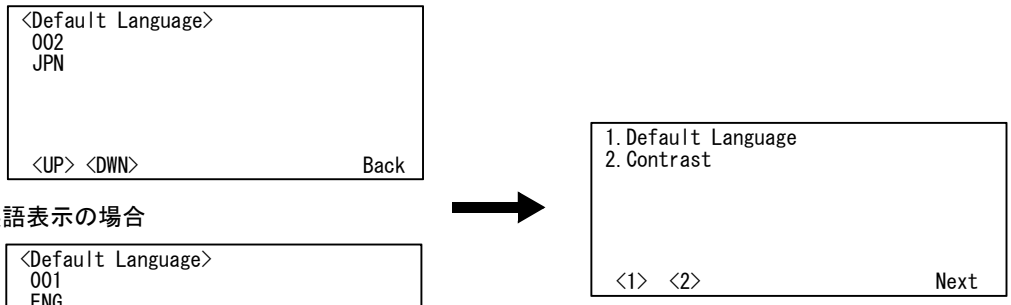
3) [F1] または [F2] キーを押して、日本語表示にする場合は画面に "JPN" を、英語表示にする場合は画面に "ENG" を表示させます。



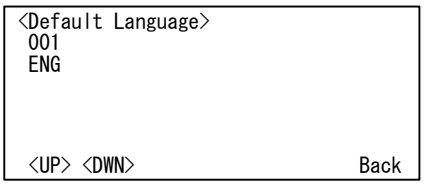
表示言語の選択 [F1]、[F2]

4) [EXE] キーを押して確定します。

日本語表示の場合

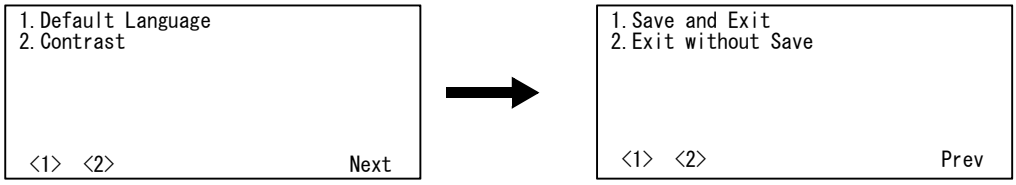


英語表示の場合



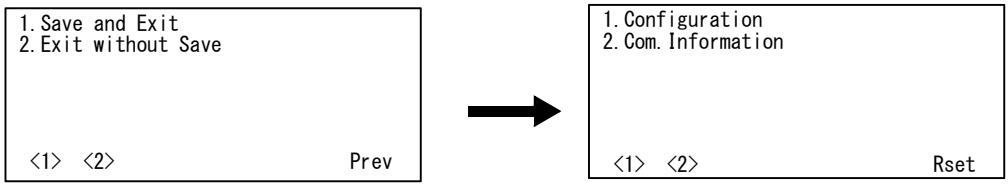
表示言語の確定 [EXE]

5) [EXE] キーを押して終了画面を表示します。



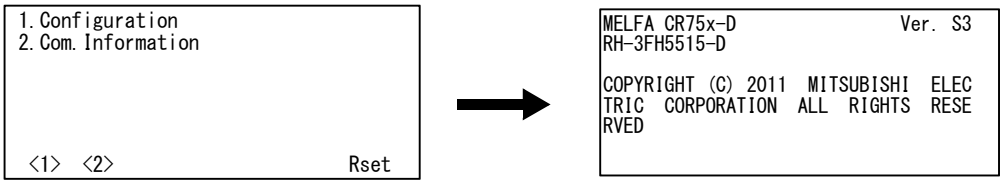
言語設定の終了 [EXE]

6) [F1] キーを押すと設定を保存します。
[F2] キーを押すと設定の保存はしません。いずれも初期設定画面に戻ります。
また、[EXE] キーを押すと設定をやり直すことができます。



設定を保存する場合 [F1]
設定を保存しない場合 [F2]
際設定する場合 [EXE]

7) [EXE] キーを押すと設定された言語で T/B が立ち上がります。

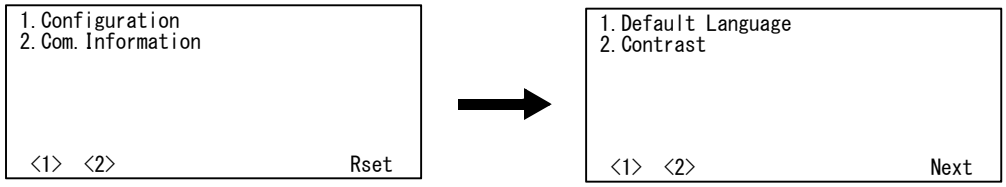


初期設定画面の終了 [EXE]

(2) コントラストの設定

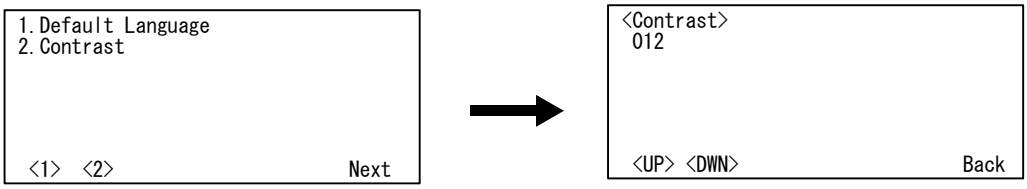
T/B の画面の明るさを 16 段階で調整できます。

1) 初期設定画面で [F1] キーを押して "1.Configuration" を選択します。



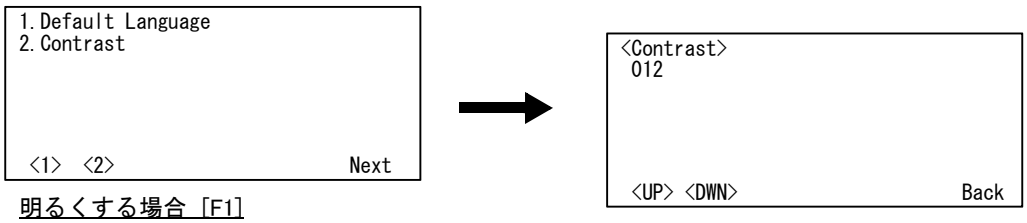
"1.Configuration" の選択 [F1]

2) [F2] キーを押して "2.Contrast" を選択します。
現在設定されている明るさが 0 から 15 の数値で表示されます。



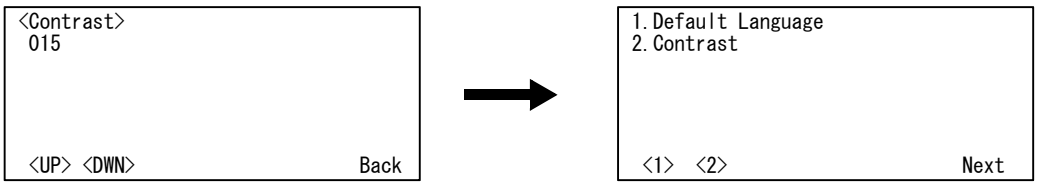
"1.Contrast" の選択 [F2]

3) 画面を明るくする場合は [F1] キーを、暗くする場合は [F2] キーを押して適切な明るさに設定します。数値が大きいほど明るくなります。



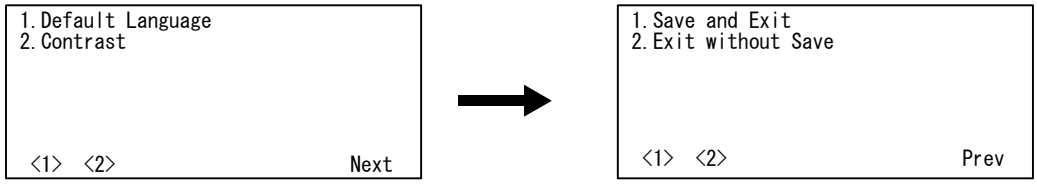
明るくする場合 [F1]
暗くする場合 [F2]

4) [EXE] キーを押して確定します。



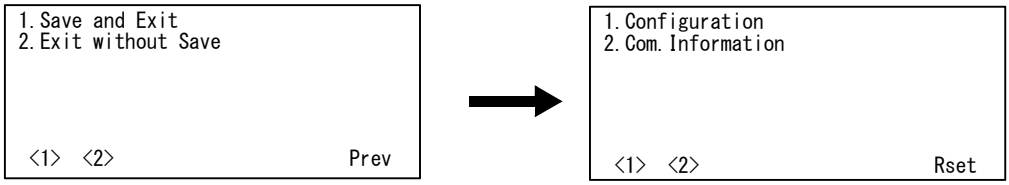
明るさの確定 [EXE]

5) [EXE] キーを押して終了画面を表示します。



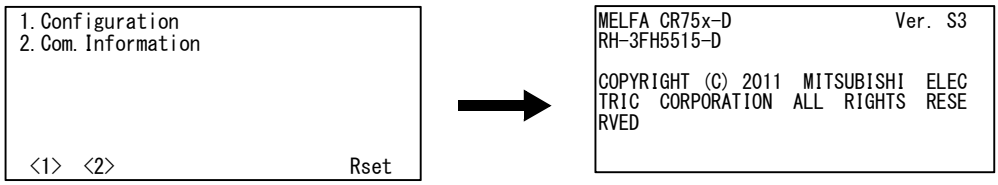
コントラスト設定の終了 [EXE]

6) [F1] キーを押すと設定を保存します。
[F2] キーを押すと設定の保存はしません。いずれも初期設定画面に戻ります。
また、[EXE] キーを押すと設定をやり直すことができます。



設定を保存する場合 [F1]
設定を保存しない場合 [F2]
際設定する場合 [EXE]

7) [EXE] キーを押すと設定された言語で T/B が立ち上がります。



初期設定画面の終了 [EXE]

4 MELFA-BASIC V

ここでは、プログラミング言語「MELFA-BASIC V」について機能及び詳細な言語仕様について説明します。

4.1 MELFA-BASIC Vの機能

ここでは、プログラミング言語「MELFA-BASIC V」の概要を説明します。ロボットの基本的な動作、信号の入出力、条件分岐方法などを解説しています。

表 4-1：解説項目一覧

No.	項目	内容	関連命令など
1	4.1.1 ロボットの動作制御	(1) 関節補間動作	Mov
2		(2) 直線補間動作	Mvs
3		(3) 円弧補間動作	Mvr, Mvr2, Mvr3, Mvc
4		(4) 連続動作	Cnt
5		(5) 加減速時間と速度制御	Accel、Oadl
6		(6) 目的位置への到達確認	Fine、Mov と Dly
7		(7) 高軌跡精度制御	Prec
8		(8) ハンド･ツール制御	HOpen、HClose、Tool
9	4.1.2 パレット演算	----------	Def Plt、Plt
10	4.1.3 プログラム制御	(1) 無条件分岐・条件分岐・待機	GoTo、If Then Else、Wait ほか
11		(2) 繰返し	For Next、While Wend
12		(3) 割込み	Def Act、Act
13		(4) サブルーチン	GoSub、Callp、On GoSub ほか
14		(5) タイマ	Dly
15		(6) 停止	End(1 サイクルの停止)、Hlt
16	4.1.4 外部信号の入出力	(1) 信号入力	M_In、M_Inb、M_Inw ほか
17		(2) 信号出力	M_Out、M_Outb、M_Outw ほか
18	4.1.5 通信	----------	Open、Close、Print、Input ほか
19	4.1.6 式と演算	(1) 演算子一覧	+、-、*、/、<>、<、> など
20		(2) 位置データの相対演算（乗算）	P1*P2
21		(3) 位置データの相対演算（加算）	P1+P2
22	4.1.7 付随節	----------	Wth、WthIf

各命令の詳細な解説は 164 ページの「4.13 命令の詳細説明」を参照願います。

4.1.1 ロボットの動作制御

(1) 関節補間動作

指定した位置へ、各関節軸単位の補間で移動します。(各関節軸単位で補間するので、先端の軌跡は直線になりません。)

■命令語

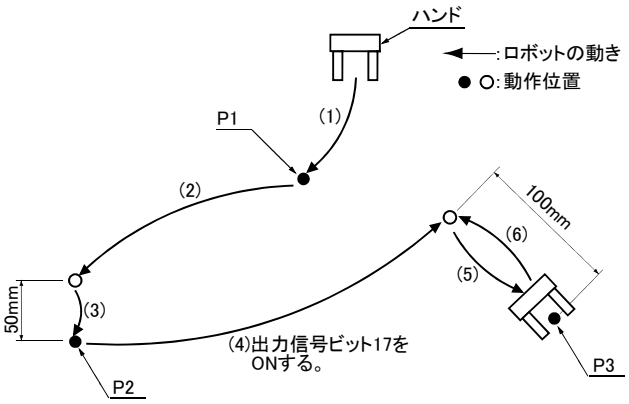
命令語	説 明
Mov	指定位置へ関節補間で移動します。Type にて補間形式が指定できます。 また Wth、WthIf の付随節が指定できます。

■文例

文例	説 明
Mov P1	'P1 へ移動します。
Mov P1+P2	'P1 と P2 の各座標成分同士を加算した位置へ移動します。124 ページ参照。
Mov P1*P2	'P1 から P2 への相対変換した位置へ移動します。124 ページ参照。
Mov P1, -50 (注)	'P1 からハンド方向に 50mm 後退した位置へ移動します。
Mov P1 Wth M_Out(17)=1	'P1 への移動開始と同時に出力信号ビット 17 を ON します。
Mov P1 WthIf M_In(20)=1, Skip	'P1 へ移動中に入力信号 20 が ON したら P1 への移動を中断、次ステップへ進みます。
Mov P1 Type 1, 0(初期値 : 遠回り)	' 各軸の動作角度が 180° を越えた場合に遠回り (または近回り) の指定をします。

■プログラム例

・ロボットの動作



注意

(注)ハンドの前進 / 後退の指定

文例、プログラムの例は垂直 6 軸ロボットの場合を示しています。

ハンドの前進 / 後退の方向は、機種毎に定められたツール座標の Z 軸方向 (+/- 方向) に依存します。

別冊の「ロボット本体セットアップから保守まで」の「動作の確認」に示したツール座標系を参照して、正しい方向を指定してください。

・プログラムの例

プログラム	説明
1 Mov P1	'(1)P1へ移動します。
2 Mov P2, -50 (注)	'(2)P2からハンド方向に50mm後退した位置へ移動します。
3 Mov P2	'(3)P2へ移動します。
4 Mov P3, -100 Wth M_Out(17)=1	'(4)P3からハンド方向に100mm後退した位置へ移動を開始し、同時に出力信号17をONします。
5 Mov P3	'(5)P3へ移動します。
6 Mov P3, -100 (注)	'(6)P3からハンド方向に100mm後退した位置へ戻ります。
7 End	'プログラムを終了します。

■関連機能

機能	説明箇所
動作速度を指定する。	105 ページの「(5) 加減速時間と速度制御」
加減速時間を指定する。	105 ページの「(5) 加減速時間と速度制御」
目的位置への到達を確認する。	107 ページの「(6) 目的位置への到達確認」
目的位置で止まらずに、次の目的位置へ連続して動作する。	104 ページの「(4) 連続動作」
直線で動作させる。	101 ページの「(2) 直線補間動作」
円や円弧を描いて動作させる。	102 ページの「(3) 円弧補間動作」
移動命令に処理を付加する。	280 ページの「Wth (ウィズ)」

(2) 直線補間動作

指定した位置へ、ハンド先端を直線補間で移動させます。

■命令語

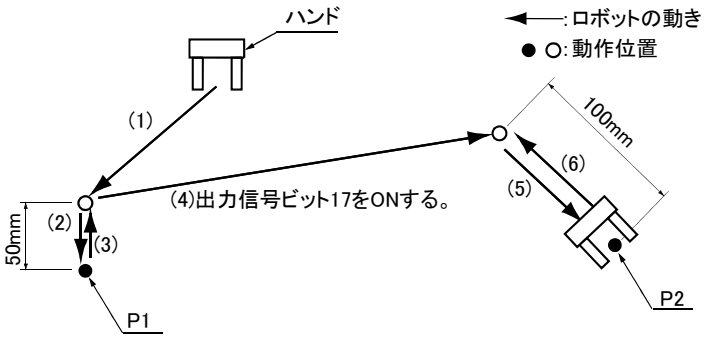
命令語	説 明
Mvs	指定位置へ直線補間で動作します。Type にて補間形式が指定できます。 Wth、WthIf の付随節が指定できます。

■文例

文例	説 明
Mvs P1	'P1 へ移動します。
Mvs P1+P2	'P1 と P2 の各座標成分同士を加算した位置へ移動します。124 ページ参照。
Mvs P1*P2	'P1 から P2 への相対変換した位置へ移動します。124 ページ参照。
Mvs P1, -50 (注)	'P1 からハンド方向に 50mm 後退した位置へ移動します。
Mvs , -50 (注)	' 現在位置からハンド方向に 50mm 後退した位置へ移動します。
Mvs P1 Wth M_Out(17)=1	'P1 への移動開始と同時に出力信号ビット 17 を ON します。
Mvs P1 WthIf M_In(20)=1, Skip	'P1 への移動中に入力信号 20 が ON したら P1 への移動を中止し、次のステップへ進みます。
Mvs P1 Type 0, 0	'P1 へ等価回転で移動します。
Mvs P1 Type 0, 1	'P1 へ 3 軸直交補間で移動します。

■プログラム例

・ロボットの動作



注意 (注)ハンドの前進 / 後退の指定

文例、プログラムの例は垂直 6 軸ロボットの場合を示しています。

ハンドの前進 / 後退の方向は、機種毎に定められたツール座標の Z 軸方向 (+/- 方向) に依存します。

別冊の「ロボット本体セットアップから保守まで」の「動作の確認」に示したツール座標系を参照して、正しい方向を指定してください。

・プログラムの例

	プログラム	説明
1	Mvs P1, -50 (注)	'(1)P1からハンド方向に50mm後退した位置へ直線補間で移動します。
2	Mvs P1	'(2)P1へ直線補間で移動します。
3	Mvs , -50 (注)	'(3)現在位置(P1)からハンド方向に50mm後退した位置へ直線補間で移動します。
4	Mvs P2, -100 Wth M_Out(17)=1 (注)	'(4)移動を開始するとともに、出力信号ビット17をONします。
5	Mvs P2	'(5)P2へ直線補間で移動します。
6	Mvs , -100 (注)	'(6)現在位置(P2)からハンド方向に100mm後退した位置へ直線補間で移動します。
7	End	'プログラムを終了します。

■関連機能

機能	説明箇所
動作速度を指定する。	105 ページの「(5) 加減速時間と速度制御」
加減速時間を指定する。	105 ページの「(5) 加減速時間と速度制御」
目的位置への到達を確認する。	107 ページの「(6) 目的位置への到達確認」
目的位置で止まらずに、次の目的位置へ連続して動作する。	104 ページの「(4) 連続動作」
関節補間で動作させる。	100 ページの「(1) 関節補間動作」
円や円弧を描いて動作させる。	102 ページの「(3) 円弧補間動作」
移動命令に処理を付加する。	280 ページの「Wth (ウィズ)」

(3) 円弧補間動作

3 点で指定した円弧上を，三次元円弧補間で移動します。
現在位置が円弧開始始点から離れている場合は始点まで直線動作してから円弧動作をします。

■命令語

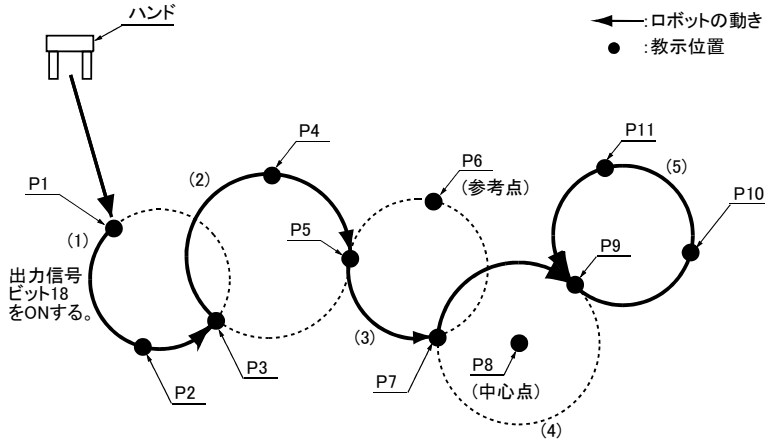
命令語	説 明
Mvr	始点、通過点、終点を指定して、始点→通過点→終点の順に円弧補間で移動します。Type にて補間形式が指定できます。Wth、WthIf の付随節が指定できます。
Mvr2	始点、終点、参考点を指定して、始点→終点まで参考点を通らない方向に円弧補間で移動します。Type にて補間形式が指定できます。Wth、WthIf の付随節が指定できます。
Mvr3	始点、終点、中心点を指定して、始点→終点までを円弧補間で移動します。 始点から終点までの扇角は 0 度＜扇角＜180 度です。 Type にて補間形式が指定できます。Wth、WthIf の付随節が指定できます。
Mvc	始点（終点）、通過点 1、通過点 2 を指定して、始点→通過点 1 →通過点 2 →終点の順に円弧補間で円周移動します。Wth、WthIf の付随節が指定できます。

■文例

文例	説 明
Mvr P1, P2, P3	'P1 → P2 → P3 を円弧補間で移動します。
Mvr P1, P2, P3 Wth M_Out(17)=1	'P1 → P2 → P3 の円弧補間動作開始とともに，出力信号ビット 17 を ON します。
Mvr P1, P2, P3 WthIf M_In(20)=1, Skip	'P1 → P2 → P3 の円弧補間動作中に入力信号ビット 20 が ON したら P1 への円弧補間動作を中止し、次のステップへ進みます。
Mvr P1, P2, P3 Type 0, 1	'P1 → P2 → P3 を円弧補間で移動します。
Mvr2 P1, P3, P11	'P1 → P3 を P11 を通らない方向で円弧補間移動します。P11 は参考点です。
Mvr3 P1, P3, P10	'P1 → P3 を扇角の小さい方向で円弧補間移動します。P10 は中心点です。
Mvc P1, P2, P3	'P1 → P2 → P3 → P1 を円周移動します。

■プログラム例

・ロボットの動作



・プログラムの例

プログラム	説明
1 Mvr P1, P2, P3 Wth M_Out(18)=1	'(1)P1→P2→P3を円弧で動作します。動作前のロボットの現在位置が始点から離れているので，最初に始点まで直線動作します。円弧動作開始と同時に(P1)出力信号ビット18をONします。
2 Mvr P3, P4, P5	'(2)P3→P4→P5を円弧で動作します。
3 Mvr2 P5, P7, P6	'(3)始点(P5)，参考点(P6)，終点 (P7)で指定された円周上で，始点から終点まで参考点を通らない方向を円弧で動作します。
4 Mvr3 P7, P9, P8	'(4)中心点(P8)，始点(P7)，終点(P9)で指定された円周上の，始点から終点までを円弧で動作します。
5 Mvc P9, P10, P11	'(5)P9→P10→P11→P9を円弧で動作します。(1周動作です。)
6 End	'プログラムを終了します。

■関連機能

機能	説明箇所
動作速度を指定する。	105 ページの「(5) 加減速時間と速度制御」
加減速時間を指定する。	105 ページの「(5) 加減速時間と速度制御」
目的位置への到達を確認する。	107 ページの「(6) 目的位置への到達確認」
目的位置で止まらずに、次の目的位置へ連続して動作する。	104 ページの「(4) 連続動作」
関節補間で動作させる。	100 ページの「(1) 関節補間動作」
直線で動作させる。	101 ページの「(2) 直線補間動作」
移動命令に処理を付加する。	280 ページの「Wth (ウィズ)」

(4) 連続動作

動作位置毎に停止することなく，複数の動作位置を連続して移動します。
命令文で，連続動作の開始と終了を指定します。連続動作中でも速度の変更ができます。

■命令語

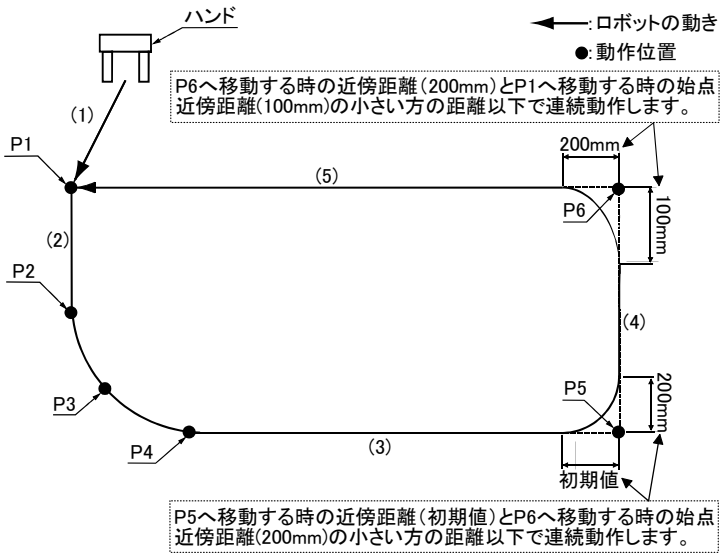
命令語	説 明
Cnt	連続動作の開始と終了を指定します。

■文例

文例	説 明
Cnt 1	’連続動作の開始を指定します。始点近傍距離は減速開始位置、終点近傍開始位置は加速終了位置です（機種ごとにより異なります）。
Cnt 1, 100, 200	’連続動作の開始指定と、その始点近傍距離 =100mm、終点近傍距離 =200mm とすることを指定します。
Cnt 0	’連続動作の終了を指定します。

■プログラム例

・ロボットの動き



注意 （注）ハンドの前進 / 後退の指定
文例、プログラムの例は垂直 6 軸ロボットの場合を示しています。
ハンドの前進 / 後退の方向は、機種毎に定められたツール座標の Z 軸方向 (+/- 方向) に依存します。
別冊の「ロボット本体セットアップから保守まで」の「動作の確認」に示したツール座標系を参照して、正しい方向を指定してください。

注意 ロボットは、その時の指定速度によって異なる軌跡で動作する場合があります。
特に、コーナー部分は内回りの軌跡を取る場合がありますので、自動運転を始める時は最初、低速で動作させ、周辺装置との干渉に注意しながら徐々に速度を上げてください。

・プログラムの例

	プログラム	説明
1	Mov P1	’(1)P1へ関節補間で移動します。
2	Cnt 1	’連続動作を有効にします。(これ以降の移動は連続動作になります。)
3	Mvr P2, P3, P4	’(2)P2まで直線動作し，連続してP4まで円弧動作をします。
4	Mvs P5	’円弧動作に連続して，P5へ直線動作します。
5	Cnt 1, 200, 100	’(3)連続動作の始点近傍距離を200mmに，終点近傍距離を100mmに設定します。
6	Mvs P6	’(4)前のP5への移動に連続して，P6へ直線動作します。
7	Mvs P1	’(5)連続して，P1へ直線動作します。
8	Cnt 0	’連続動作を無効にします。
9	End	’プログラムを終了します。

■関連機能

機能	説明箇所
動作速度を指定する。	105 ページの「(5) 加減速時間と速度制御」
加減速時間を指定する。	105 ページの「(5) 加減速時間と速度制御」
目的位置への到達を確認する。	107 ページの「(6) 目的位置への到達確認」
関節補間で動作させる。	100 ページの「(1) 関節補間動作」
直線で動作させる。	101 ページの「(2) 直線補間動作」
円や円弧を描いて動作させる。	102 ページの「(3) 円弧補間動作」

(5) 加減速時間と速度制御

加減速時の最高加減速度に対する割合、および動作速度を指定できます。

■命令語

命令語	説 明
Accel	移動時の加速度と、減速度を最高加減速度に対する割合(%)で指定します。
Ovrd	プログラム全体にかかる動作速度を、最高速度に対する割合(%)で指定します。
JOvrd	関節補間動作時の速度を、最高速度に対する割合(%)で指定します。
Spd	直線、円弧補間動作時の速度を、ハンド先端速度(mm/s)で指定します。
Oadl	最適加減速度機能を有効/無効の指定します。

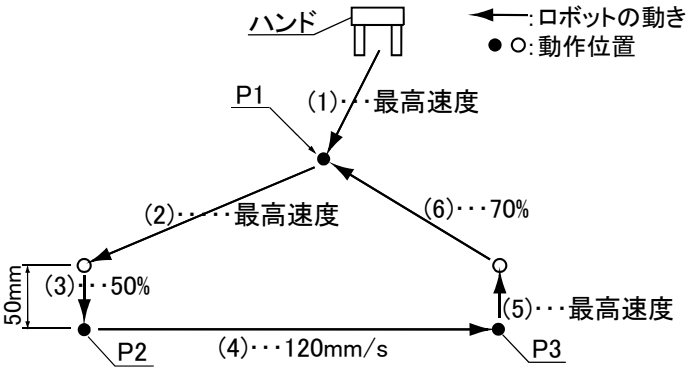
■文例

文例	説 明
Accel	’加速度とも100%に設定します。
Accel 60, 80	’加速度60%, 減速度80%に設定します。 (最高加減速時間が0.2秒の場合 加速時間 0.2÷0.6=0.33、 減速時間 0.2÷0.8=0.25秒になります。)
Ovrd 50	’関節補間, 直線補間, 円弧補間動作とも最高速度の50%に設定します。
JOvrd 70	’関節補間動作を最高速度の70%に設定します。
Spd 30	’直線補間, 円弧補間動作時の速度を30mm/sに設定します。
Oadl ON	’最適加減速機能を有効にします。

・関節補間時の動作速度 ..... コントローラ(T/B)の設定値 ×Ovrd 命令の設定値 ×JOvrd 命令の設定値
・直線、円弧補間時の動作速度..... コントローラ(T/B)の設定値 ×Ovrd 命令の設定値 ×Spd 命令の設定値

■プログラム例

・ロボットの動作



注意

（注）ハンドの前進/後退の指定
文例、プログラムの例は垂直6軸ロボットの場合を示しています。
ハンドの前進/後退の方向は、機種毎に定められたツール座標のZ軸方向(+/−方向)に依存します。
別冊の「ロボット本体セットアップから保守まで」の「動作の確認」に示したツール座標系を参照して、正しい方向を指定してください。

・プログラムの例

プログラム		説明
1	Ovrd 100	’全体にかかる動作速度を最大に設定します。
2	Mvs P1	’(1)P1へ最高速度で移動します。
3	Mvs P2, -50 (注)	’(2)P2からハンド方向に50mm後退した位置へ最高速度で移動します。
4	Ovrd 50	’全体にかかる動作速度を最高速度の半分に設定します。
5	Mvs P2	’(3)P2へ初期設定速度の半分で直線動作します。
6	Spd 120	’先端速度を120mm/sに設定します。(オーバーライドが50%なので実際は60mm/sで動作します。)
7	Ovrd 100	’実際の先端速度を120mm/sにするために、動作速度の割合を100%に設定します。
8	Accel 70, 70	’加減速度とも最高加減速度の70%に設定します。
9	Mvs P3	’(4)P3へ先端速度120mm/sで直線動作します。
10	Spd M_NSpd	’先端速度を初期値に戻します。
11	JOvrd 70	’関節補間動作時の速度を70%に設定します。

プログラム	説明
12 Accel	’加減速度とも最高加減速度に戻します。
13 Mvs , -50（注）	’(5)現在位置(P3)からハンド方向に50mm後退した位置へ直線動作時の初期設定速度で直線移動します。
14 Mvs P1	’(6)最高速度の70%でP1へ移動します。
15 End	’プログラムを終了します。

■関連機能

機能	説明箇所
関節補間で動作させる。	100 ページの「(1) 関節補間動作」
直線で動作させる。	101 ページの「(2) 直線補間動作」
円や円弧を描いて動作させる。	102 ページの「(3) 円弧補間動作」
目的位置で止まらずに、次の目的位置へ連続して動作する。	104 ページの「(4) 連続動作」

(6) 目的位置への到達確認

位置決め完了条件をパルス数で指定できます。(Fine 命令) 連続動作時は本指定は無効となります。

■命令語

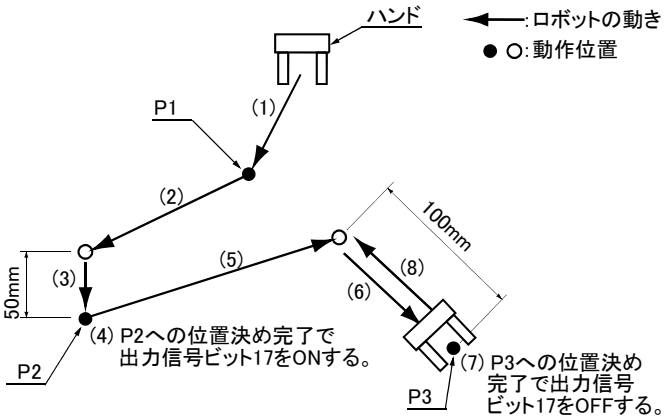
命令語	説 明
Fine	位置決め完了条件をパルス数で指定します。指定したパルス数を小さくするとより正確な位置決めが可能になります。
Mov と Dly	Mov の動作指令後に Dly 命令 (タイマ) で位置決め完了にする。 (ベルト駆動方式のロボットに有効です。)

■文例

文例	説 明
Fine 100	' 位置決め完了条件を 100 パルスに設定します。
Mov P1	'P1 へ関節補間で移動します。(指令値レベルで完了します。)
Dly 0.5	' 動作命令後の位置決めをタイマによりおこないます。 (ベルト駆動方式のロボットに有効です。)

■プログラム例

・ロボットの動作



注意 (注) ハンドの前進 / 後退の指定

文例、プログラムの例は垂直 6 軸ロボットの場合を示しています。

ハンドの前進 / 後退の方向は、機種毎に定められたツール座標の Z 軸方向 (+/- 方向) に依存します。

別冊の「ロボット本体セットアップから保守まで」の「動作の確認」に示したツール座標系を参照して、正しい方向を指定してください。

・プログラムの例

プログラム	説明
1 Cnt 0	' Fine命令はCnt命令OFF中のみ有効です。
2 Mvs P1	' (1)P1へ関節補間で移動します。
3 Mvs P2, -50 (注)	' (2)P2からハンド方向に50mm後退した位置へ関節補間で移動します。
4 Fine 50	' 位置決め完了パルスを50に設定します。
5 Mvs P2	' (3)P2へ直線補間で移動します。(位置決め完了パルスが50以下でMvsが終了します)
6 M_Out(17)=1	' (4)位置決めパルスが50パルスになったら出力信号17をONします。
7 Fine 1000	' 位置決め完了パルスを1000に設定します。
8 Mvs P3, -100 (注)	' (5)P3からハンド方向に100mm後退した位置へ直線で移動します。
9 Mvs P3	' (6)P3へ直線で移動します。
10 Dly 0.1	' 位置決めをタイマによりおこないます。
11 M_Out(17)=0	' (7)出力信号17をOFFします。
12 Mvs , -100 (注)	' (8)現在位置(P3)からハンド方向に100mm後退した位置へ直線で移動します。
13 End	' プログラムを終了します。

■関連機能

機能	説明箇所
関節補間で動作させる。	100 ページの「(1) 関節補間動作」
直線で動作させる。	101 ページの「(2) 直線補間動作」
目的位置で止まらずに、次の目的位置へ連続して動作する。	104 ページの「(4) 連続動作」

(7) 高軌跡精度制御

ロボットの動作軌跡を向上させて動作させることができます。

■命令語

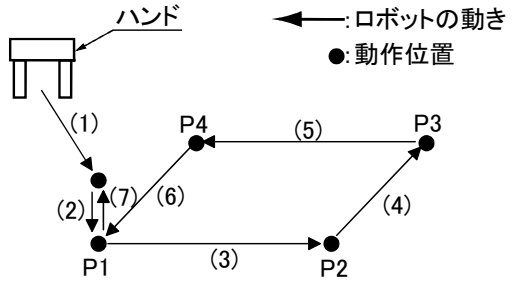
命令語	説 明
Prec	高精度モードの有効、無効を指定します。

■文例

文例	説 明
Prec On	’高精度モードを有効にする。
Prec Off	’高精度モードを無効にする。

■プログラム例

・ロボットの動作



⚠注意 （注）ハンドの前進／後退の指定

文例、プログラムの例は垂直 6 軸ロボットの場合を示しています。

ハンドの前進／後退の方向は、機種毎に定められたツール座標の Z 軸方向 (+/− 方向) に依存します。

別冊の「ロボット本体セットアップから保守まで」の「動作の確認」に示したツール座標系を参照して、正しい方向を指定してください。

・プログラムの例

	プログラム	説明
1	Mov P1, −50 （注）	’(1)P1からハンド方向に50mm後退した位置へ関節補間で移動します。
2	Ovrd 50	’ 動作速度を最高速度の半分に設定します。
3	Mvs P1	’(2)P1へ直線で移動します。
4	Prec On	’ 高軌跡モードを有効にします。
5	Mvs P2	’(3)P1からP2へ高軌跡精度で移動します。
6	Mvs P3	’(4)P2からP3へ高軌跡精度で移動します。
7	Mvs P4	’(5)P3からP4へ高軌跡精度で移動します。
8	Mvs P1	’(6)P4からP1へ高軌跡精度で移動します。
9	Prec Off	’ 高軌跡モードを無効にします。
10	Mvs P1, −50	’(7)P1からハンド方向に50mm後退した位置へ直線補間で戻ります。
11	End	’プログラムを終了します。

⚠注意

Prec 命令はロボットの先端軌跡精度は向上しますが動作の加減速度をさげているためタクト時間は延びる可能性があります。

Cnt 命令は入れない方が精度はあがります。ただし先端の速度は保証できません。

(8) ハンド･ツール制御

ハンドの開閉，ツール形状の指定ができます。

■命令語

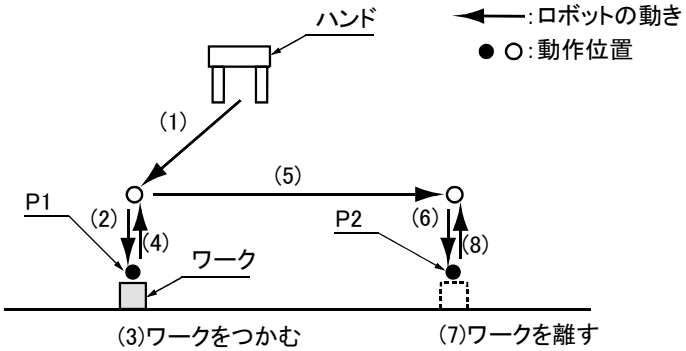
命令語	説 明
HOpen	指定したハンドを開きます。
HClose	指定したハンドを閉じます。
Tool	使用ツールの形状を設定し、制御点を合わせます。

■文例

文例	説 明
HOpen 1	’1番のハンドを開きます。
HOpen 2	’2番のハンドを開きます。
HClose 1	’1番のハンドを閉じます。
HClose 2	’2番のハンドを閉じます。
Tool (0, 0, 95, 0, 0, 0)	’ロボットの制御点の位置をフランジ面からの延長方向に95mmのところにします。

■プログラム例

・ロボットの動作



注意 （注）ハンドの前進 / 後退の指定
文例、プログラムの例は垂直6軸ロボットの場合を示しています。
ハンドの前進 / 後退の方向は、機種毎に定められたツール座標のZ軸方向(+/- 方向)に依存します。
別冊の「ロボット本体セットアップから保守まで」の「動作の確認」に示したツール座標系を参照して、正しい方向を指定してください。

・プログラムの例

	プログラム	説明
1	Tool (0, 0, 95, 0, 0, 0)	’ハンド長を95mmに設定します。
2	Mvs P1, -50（注）	’(1)P1からハンド方向に50mm後退した位置へ関節補間で移動します。
3	Ovrd 50	’動作速度を最高速度の半分に設定します。
4	Mvs P1	’(2)P1へ直線で移動します。(ワークをつかみに行きます)
5	Dly 0.5	’目的位置到達完了のため、0.5秒待ちます。
6	HClose 1	’(3)ハンド1を閉じます。(ワークをつかみます)
7	Dly 0.5	’0.5秒待ちます。
8	Ovrd 100	’動作速度を最大に設定します。
9	Mvs , -50（注）	’(4)現在位置(P1)からハンド方向に50mm後退した位置へ直線で移動します。(ワークを持ち上げます)
10	Mvs P2, -50（注）	’(5)P2からハンド方向に50mm後退した位置へ関節補間で移動します。
11	Ovrd 50	’動作速度を最高速度の半分に設定します。
12	Mvs P2	’(6)P2へ直線で移動します。(ワークを置きに行きます)
13	Dly 0.5	’目的位置到達完了のため、0.5秒待ちます。
14	HOpen 1	’(7)ハンド1を開きます。(ワークを離します)
15	Dly 0.5	’0.5秒待ちます。
16	Ovrd 100	’動作速度を最大に設定します。
17	Mvs , -50（注）	’(8)現在位置(P2)からハンド方向に50mm後退した位置へ直線で移動します。(ワークから離れます)
18	End	’プログラムを終了します。

■関連機能

機能	説明箇所
付随節	280ページの「Wth(ウィズ)」

4.1.2 パレット演算

規則正しくワークを並べる作業(パレタイズ)や、規則正しく並んだワークを取り出す作業(デパレタイズ)の場合、パレット機能を使えば基準となるワークの位置だけを教示して残りの位置は演算で求めることができます。

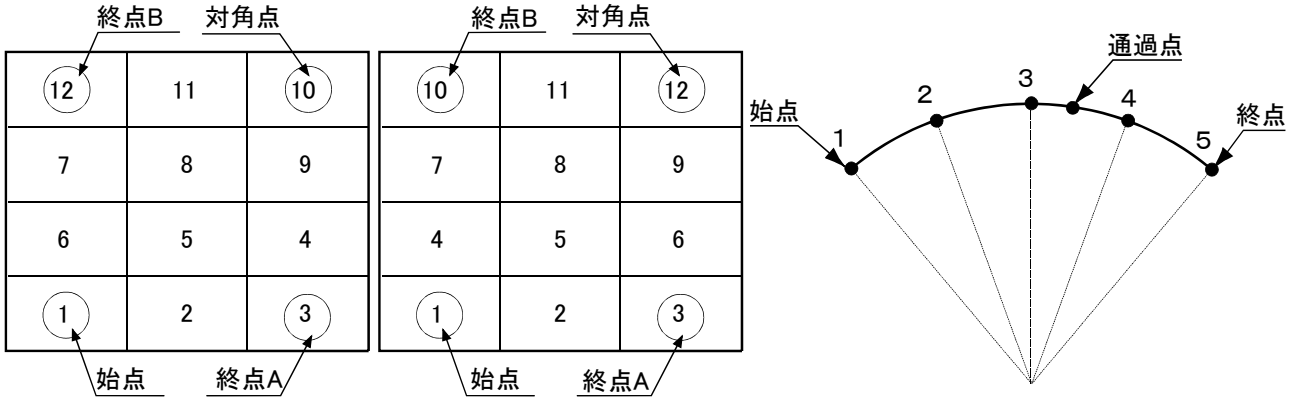
■命令語

命令語	説 明
Def Plt	使用するパレットを定義します。
Plt	パレット上の指定した位置を演算で求めます。

■文例

文例	説 明
Def Plt 1, P1, P2, P3, P4, 4, 3, 1	’パレット番号 1 を、始点 =P1、終点 A=P2、終点 B=P3、対角点 =P4 の 4 点で指定される場所と大きさの中に、個数 A=4、個数 B=3 の計 12 個(4×3)の作業位置があり、パレットパターン =1(ｼﾞｸﾞｻﾞｸﾞ)で演算して使用することを定義します。
Def Plt 2, P1, P2, P3, , 8, 5, 2	’パレット番号 2 を、始点 =P1、終点 A=P2、終点 B=P3 の 3 点で指定される場所と大きさの中に、個数 A=8、個数 B=5 の計 40 個(5×8)の作業位置があり、パレットパターン =2(同一方向)で演算して使用することを定義します。
Def Plt 3, P1, P2, P3, , 5, 1, 3	’パレット番号 3 を、始点 =P1、通過点 =P2、終点 =P3 の 3 点で指定される円弧上に、計 5 個の作業位置がある、円弧パレットを使用することを定義します。
(Plt 1, 5)	’パレット番号 1 の 5 番目の位置を演算します。
(Plt 1, M1)	’数値変数 M1 の値が示す、パレット番号 1 内の位置を演算します。

位置の指定とパレットパターンの関係を以下に示します。



ジグザグ

パレットパターン =1(ジグザグ)(姿勢等分割)

パレットパターン =11(ジグザグ)(姿勢固定)

同一方向

パレットパターン =2(同一方向)(姿勢等分割)

パレットパターン =12(同一方向)(姿勢固定)

円弧パレット

パレットパターン =3(円弧パレット)(姿勢等分割)

パレットパターン =13(円弧パレット)(姿勢固定)

＜パレット定義をする位置データの姿勢に関する注意事項＞

⚠注意　姿勢成分（A，B，C）が ±180 度近辺の位置データをパレット定義の＜始点＞、＜終点 A，B＞、＜対角点＞とする場合は、以下に示します「■解説」をご確認の上、ご使用願います。

■解説

姿勢成分（A，B，C）が 180 度となる位置では、同じ姿勢であっても成分の値は +180 度、または −180 度のいずれかとなります。これは内部的な演算誤差によるもので、どちらの値になるかの規則性はありません。

この位置をパレット定義の＜始点＞、＜終点＞、＜対角点＞とする場合、同一の姿勢成分に +180 度と −180 度が混在していると、パレットの格子点の位置は −180 度 ～ +180 度の間を分割して算出されるため、ハンドが回転し予期せぬ動作となってしまいます。

姿勢成分の +180 度と −180 度は同じ姿勢となりますので、パレットの定義に使用する位置データは、符号を + または − のいずれかに揃えてから使用してください。

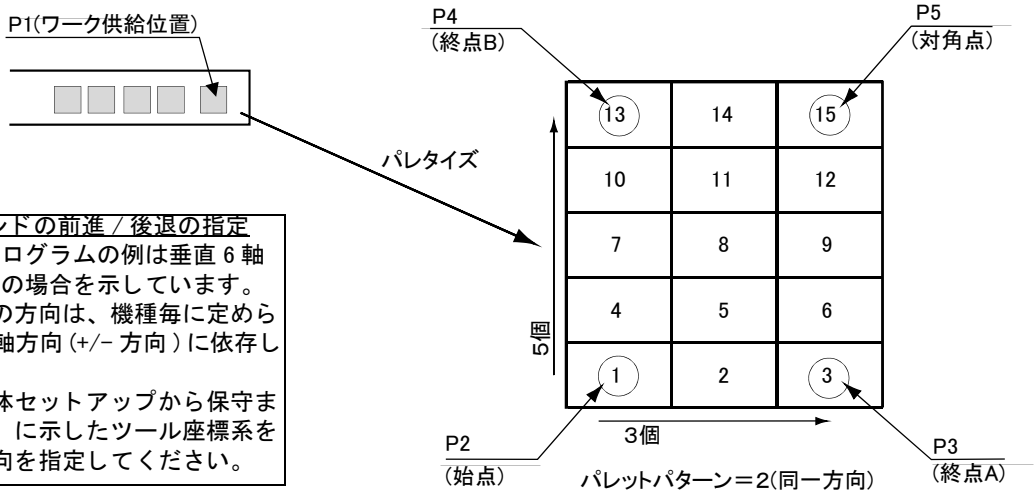
また、姿勢成分が ±180 度近辺（例えば、+179 度と −179 度）の場合も、符号が混在していると同様の現象となります。その場合は、姿勢成分に 360 度を加算もしくは減算して、符号が揃うように値を補正してください。（例えば、−179 度を + の符号に揃える場合は、360 度を加算した +181 度に補正します。）

「・プログラムの例 1」では、パレットの全格子点においてハンドの向きが同じである場合に（A，B，C 軸の値が同じ）、終点（P3、P4）と対角点（P5）の姿勢成分を始点（P2）に合わせる例（行番 10 ～ 90 番）と、「・プログラムの例 2」では、パレット定義位置の姿勢成分が ±180 度近辺で、終点（P3、P4）と対角点（P5）の C 軸の値が −178 度より小さい、または +178 度より大きい場合、始点（P2）と同じ符号に値を補正する例（行番 10 ～ 100）を示します。（±178 度を補正のしきい値としています）パレットの精度があまり良くなく、ハンドの向きをわずかに補正しなければならないような場合の参考としてください。

また、パレットパターンに 11 ～ 13 を指定するとパレット演算で求めた位置変数の姿勢データは＜始点＞の姿勢が代入され、ハンドの向きを固定することもできます。（1 ～ 3 を指定した場合の姿勢データは、＜始点＞−＜終点＞間を等分割したものが代入されます）

■プログラム例

・ロボットの動作



⚠注意　（注）ハンドの前進 / 後退の指定
文例、プログラムの例は垂直 6 軸ロボットの場合を示しています。
ハンドの前進 / 後退の方向は、機種毎に定められたツール座標の Z 軸方向（+/− 方向）に依存します。
別冊の「ロボット本体セットアップから保守まで」の「動作の確認」に示したツール座標系を参照して、正しい方向を指定してください。

⚠注意　パレット演算(Plt 命令)により算出される格子点の構造フラグ(位置データの Fl1)は、パレット定義の始点の値が採用されます。
したがって、パレット定義の各点で構造フラグの異なる位置データを使用すると、所望のパレット動作とは異なる動きとなってしまいます。
パレット定義の始点、終点 A，B、対角点には、構造フラグの値が全て同じとなる位置データを使用してください。また、格子点の多回転フラグ(位置データの Fl2)についても、パレット定義の始点の値が採用されます。パレット定義の各点で多回転フラグの異なる位置データを使用すると、パレット動作で経由するロボットの位置や補間命令の種類(関節補間、直線補間等)によっては、ハンドが回転し予期せぬ動作となってしまいます。そのような場合は、補間命令の引数 Type を用いて、姿勢の遠回り / 近回り動作を適切に設定し、ハンドが所望の動作となるよう調整してください。

・プログラムの例 1
パレットの全格子点においてハンドの向きが同じ場合(A，B，C 軸の値が同じ)

	プログラム	説明
1	P3.A=P2.A	’P3の姿勢成分(A)にP2の姿勢成分(A)を代入します。
2	P3.B=P2.B	’P3の姿勢成分(B)にP2の姿勢成分(B)を代入します。
3	P3.C=P2.C	’P3の姿勢成分(C)にP2の姿勢成分(C)を代入します。
4	P4.A=P2.A	’P4の姿勢成分(A)にP2の姿勢成分(A)を代入します。
5	P4.B=P2.B	’P4の姿勢成分(B)にP2の姿勢成分(B)を代入します。
6	P4.C=P2.C	’P4の姿勢成分(C)にP2の姿勢成分(C)を代入します。
7	P5.A=P2.A	’P5の姿勢成分(A)にP2の姿勢成分(A)を代入します。
8	P5.B=P2.B	’P5の姿勢成分(B)にP2の姿勢成分(B)を代入します。
9	P5.C=P2.C	’P5の姿勢成分(C)にP2の姿勢成分(C)を代入します。
10	Def Plt 1, P2, P3, P4, P5, 3, 5, 2	’パレットを定義します。 パレット番号=1、始点=P2、終点A=P3、終点B=P4、対角点=P5、 個数A=3、個数B=5、パレットパターン=2(同一方向)
11	M1=1	’数値変数M1に値1を代入します。(M1はカウンタに使用します)
12	*LOOP	’ジャンプ先として，ラベルLOOPを指定します。
13	Mov P1, -50（注）	’P1からハンド方向に50mm後退した位置へ関節補間で移動します。
14	Ovrd 50	’動作速度を最高速度の半分に設定します。
15	Mvs P1	’P1へ直線で移動します。(ワークをつかみに行きます)
16	HClose 1	’ハンド1を閉じます。(ワークをつかみます)
17	Dly 0.5	’0.5秒待ちます。
18	Ovrd 100	’動作速度を最大に設定します。
19	Mvs , -50（注）	’現在位置(P1)からハンド方向に50mm後退した位置へ直線で移動します。 (ワークを持ち上げます)
20	P10=(Plt 1, M1)	’数値変数M1の値が示す、パレット番号1内の位置を演算し，結果をP10に代入します。
21	Mov P10, -50（注）	’P10からハンド方向に50mm後退した位置へ関節補間で移動します。
22	Ovrd 50	’動作速度を最高速度の半分に設定します。
23	Mvs P10	’P10へ直線で移動します。(ワークを置きに行きます)
24	HOpen 1	’ハンド1を開します。(ワークを離します)
25	Dly 0.5	’0.5秒待ちます。
26	Ovrd 100	’動作速度を最大に設定します。
27	Mvs , -50（注）	’現在位置(P10)からハンド方向に50mm後退した位置へ直線で移動します。 (ワークから離れます)
28	M1=M1+1	’数値変数M1の値に1を足します。(パレットカウンタを進めます)
29	If M1<=15 Then *LOOP	’数値変数M1の値が15以下ならばラベルLOOPへジャンプして処理を繰り返します。 そうでなければ次の行へ進みます。
30	End	’プログラムを終了します。

・プログラムの例 2
姿勢成分が ±180 度近辺の場合の補正

プログラム	説明
1 If Deg(P2.C)<0 Then GoTo*MINUS	'P2の姿勢成分(C)の符号を確認し、-(マイナス)ならばラベルMINUSへジャンプします。
2 If Deg(P3.C)<-178 Then P3.C =P3.C+Rad(+360)	'P3の姿勢成分が-180度近辺の場合は360度を加算して+の値に補正します。
3 If Deg(P4.C)<-178 Then P4.C =P4.C+Rad(+360)	'P4の姿勢成分が-180度近辺の場合は360度を加算して+の値に補正します。
4 If Deg(P5.C)<-178 Then P5.C=P5.C+Rad(+360)	'P5の姿勢成分が-180度近辺の場合は360度を加算して+の値に補正します。
5 GoTo *DEFINE	'ラベルDEFINEの行へ無条件にジャンプします
6 *MINUS	'ジャンプ先として，ラベルMINUSを指定します。
7 If Deg(P3.C) > +178 Then P3.C =P3.C-Rad(+360)	'P3の姿勢成分が+180度近辺の場合は360度を加算しての-値に補正します。
8 If Deg(P4.C) > +178 Then P4.C =P4.C-Rad(+360)	'P4の姿勢成分が+180度近辺の場合は360度を加算しての-値に補正します。
9 If Deg(P5.C) > +178 Then P5.C =P5.C-Rad(+360)	'P5の姿勢成分が+180度近辺の場合は360度を加算しての-値に補正します。
10 *DEFINE	ジャンプ先として，ラベルDEFINEを指定します。
11 Def Plt 1, P2, P3, P4, P5, 3, 5, 2	' パレットを定義します。 パレット番号=1、始点=P2、終点A=P3、 終点B=P4、対角点=P5、個数A=3、個数B=5、 パレットパターン=2(同一方向)
12 M1=1	' 数値変数M1に値1を代入します。 (M1はカウンタに使用します)
13 *LOOP	' ジャンプ先として，ラベルLOOPを指定します。
14 Mov P1, -50 （注）	' P1からハンド方向に50mm後退した位置へ関節補間で移動します。
15 Ovrd 50	' 動作速度を最高速度の半分に設定します。
16 Mvs P1	' P1へ直線で移動します。(ワークをつかみに行きます)
17 HClose 1	' ハンド1を閉じます。(ワークをつかみます)
18 Dly 0.5	' 0.5秒待ちます。
19 Ovrd 100	' 動作速度を最大に設定します。
20 Mvs , -50 （注）	' 現在位置(P1)からハンド方向に50mm後退した位置へ直線で移動します。(ワークを持ち上げます)
21 P10=(Plt 1, M1)	' 数値変数M1の値が示す、パレット番号1内の位置を演算し，結果をP10に代入します。
22 Mov P10, -50 （注）	' P10からハンド方向に50mm後退した位置へ関節補間で移動します。
23 Ovrd 50	' 動作速度を最高速度の半分に設定します。
24 Mvs P10	' P10へ直線で移動します。(ワークを置きに行きます)
25 HOpen 1	' ハンド1を開します。(ワークを離します)
26 Dly 0.5	' 0.5秒待ちます。
27 Ovrd 100	' 動作速度を最大に設定します。
28 Mvs , -50	' 現在位置(P10)からハンド方向に50mm後退した位置へ直線で移動します。(ワークから離れます)
29 M1=M1+1	' 数値変数M1の値に1を足します。 (パレットカウンタを進めます)
30 If M1<=15 Then *LOOP	' 数値変数M1の値が15以下ならばラベルLOOPへジャンプして処理を繰り返します。そうでなければ次の行へ進みます。
31 End	'プログラムを終了します。

4.1.3 プログラム制御

分岐、割込み、タイマ、サブルーチンコール、停止などプログラムの流れを制御できます。

(1) 無条件分岐・条件分岐・待機

指定行へプログラムの流れを無条件に、または条件判断して分岐することができます。

■命令語

命令語	説 明
GoTo	指定したラベルに無条件にジャンプします。
On GoTo	指定変数の値に応じてジャンプします。値の条件は整数値順です。(1,2,3,4,.....)
If Then Else (1 行での記述)	指定条件に応じて命令を実行します。値の条件は任意に指定できます。 条件が成立した場合は Then の後ろの記述を実行、条件が不成立の場合は Else の後の記述を実行します。一行で記述します。

If Then
Else
EndIf
(複数行での記述)
	指定変数と、その値の指定条件に応じて複数行を処理します。値の条件は任意に指定できます。条件の種類は、1 命令文当たり 1 種類です。
	条件が成立した場合は Then の次行から Else までの行を実行、条件が不成立の場合は Else の後の行から EndIf までの行を実行します。

Select Case End Select	指定変数と、その値の指定条件に応じてジャンプします。値の条件は任意に指定できます。 条件の種類は、1 命令文当たり複数種類の指定が可能です。
Wait	変数が指定の値になるまで待機します。

■文例

文例	説 明
GoTo *FIN	’ラベル FIN の行へ無条件にジャンプします。
On M1 GoTo *L1, *L2, *L3	’数値変数 M1 の値が 1 ならラベル L1, 2 ならラベル 2, 3 ならラベル 3 へジャンプします。値が該当しない場合は次の行へ進みます。
If M1=1 Then *L100	’数値変数 M1 の値が 1 ならラベル *L1 へ分岐 , そうでないなら次の行へ進みます。
If M1=1 Then *L10 Else *L20	’数値変数 M1 の値が 1 ならラベル *L10 へ , そうでないならラベル *L20 へ分岐します。

If M1=1 Then
M2=1
M3=2
Else
M2=-1
M3=-2
EndIf
	’数値変数 M1 の値が 1 なら M2=1、M3=2 を実行、M1 が 1 以外なら M2=-1M3=-2 を実行します。

Select M1	’数値変数 M1 の値により該当する Case 文へ分岐します。
Case 10 : Break	’値が 10 なら Case 10 と次の Case IS 11 の間のみ実行します。
Case 11 : Break	’値が 11 なら Case11 と次の Case IS < 5 の間のみ実行します。
Case IS < 5 : Break	’値が 5 より小さいなら Case IS < 5 と次の Case 6 TO 9 の間のみ実行します。
Case 6 TO 9 : Break	’値が 6 から 9 なら Case 6 TO 9 と次の Default の間のみ実行します。
Default : Break	’値が上記のいずれにも該当しないなら Default と次の End Select の間のみ実行します。
End Select	’Select Case 文を終了します。
Wait M_In(1)=1	’入力信号ビット 1 が ON するまで待機します。

■関連機能

機能	説明箇所
繰返し	116 ページの「(2) 繰返し」
割込み	117 ページの「(3) 割込み」
サブルーチン	118 ページの「(4) サブルーチン」
外部信号の入力	121 ページの「(1) 信号入力」

(2) 繰返し

指定条件に応じて、複数の命令文を繰返し実行することができます。

■命令語

命令語	説 明
For Next	For 文と Next 文の間を、指定条件を満たすまで繰り返します。
While WEnd	While 文と WEnd 文の間を、指定条件を満たしている間繰り返します。

■文例

文例	説 明
For M1=1 To 10 : Next	' For 文と Next 文の間を 10 回繰り返します。 数値変数 M1 の値は、最初に 1 が代入され、繰り返す毎に 1 が加えられます。
For M1=0 To 10 Step 2 : Next	' For 文と Next 文の間を 6 回繰り返します。 数値変数 M1 の値は、最初に 0 が代入され、繰り返す毎に 2 が加えられます。
While (M1>=1) And (M1<=10) : WEnd	' 数値変数 M1 の値が、1 以上でかつ 10 以下の間、While 文と WEnd 文の間を繰り返します。

■関連機能

機能	説明箇所
無条件分岐・条件分岐。	114 ページの「(1) 無条件分岐・条件分岐・待機」
割込み。	117 ページの「(3) 割込み」
入力信号待ち。	121 ページの「(1) 信号入力」

(3) 割込み

指定条件が成立したら実行中の命令文を中断して、指定行へ分岐させることができます。

■命令語

命令語	説 明
Def Act	割込み条件と割込み発生時の処理を定義します。
Act	割込みの許可 / 禁止を指定します。
Return	割込み処理をサブルーチンの呼出しとした場合に、割込み元の行に戻ります。

■文例

文例	説 明
Def Act 1, M_In(10)=1 GoSub *L100	’割り込み番号 1 に入力信号ビット 10 が ON したら、ロボット減速停止後にラベル *L100 のサブルーチンを呼び出すことを定義します。 減速時間は Accel 命令や Ovrd 命令に依存します。
Def Act 2, M_In(11)=1 GoSub *L200, L	’割り込み番号 2 に入力信号ビット 11 が ON したら、実行中の命令文が完了後ラベル *L200 のサブルーチンを呼び出すことを定義します。
Def Act 3, M_In(12)=1 GoSub *L300, S	’割り込み番号 3 に入力信号ビット 12 が ON したら、ロボットを最短時間、最短距離で減速停止後ラベル *L300 のサブルーチンを呼び出すことを定義します。
Act 1=1	’優先番号 1 の割込みを許可します。
:	
Act 2=0	’優先番号 2 の割込みを禁止します。
:	
*L100	’割込みが発生した行に戻ります。
:	
Return 0	
*L200	’割込みが発生した次の行に戻ります。
:	
Return 1	
*L300	’ 割込みが発生した行に戻ります。
:	
Return 0	

■関連機能

機能	説明箇所
無条件分岐・条件分岐。	114 ページの「(1) 無条件分岐・条件分岐・待機」
サブルーチン。	118 ページの「(4) サブルーチン」
通信。	122 ページの「4.1.5 通信」

(4) サブルーチン

サブルーチンやサブプログラムを使うことができます。
本機能を使うことにより、プログラムを共用してステップ数を節約したり、プログラムを階層構造にして分かり易くすることができます。

■命令語

命令語	説 明
GoSub	指定ラベルのサブルーチンを呼び出します。
On GoSub	指定変数の値に応じたサブルーチンを呼出します。値の条件は整数値順です。(1,2,3,4,.....)
Return	GoSub 命令で呼び出した次の行へ戻ります。
CallP	指定したプログラムを呼び出します。呼び出されたプログラムの End 命令で元のプログラムの次の行へ戻ります。呼び出すプログラムへ引数としてデータを渡すことができます。
FPrm	CallP 命令で呼び出されたプログラムで引数を受け取ります。

■文例

文例	説 明
GoSub *GET	' ラベル GET からのサブルーチンを呼び出します。
On M1 GoSub *L1, *L2, *L3	' 数値変数 M1 の値が 1 ならラベル L1, 2 ならラベル L2, 3 ならラベル L3 のサブルーチンを呼出します。
:	値が該当しない場合は次の行へ進みます。
:	
:	
*L1	
:	
Return	
*L2	
:	
Return	
*L3	
:	
Return	'GoSub 命令で呼び出した次の行へ戻ります。
Return	' GoSub 命令で呼び出した次の行へ戻ります。
CallP "10"	' 10 番のプログラムを呼び出します。
CallP "20", M1, P1	' 20 番のプログラムへ数値変数 M1 と位置変数 P1 を渡して呼び出します。
FPrm M10, P10	' サブプログラムにおいて CallP 命令で渡された数値変数を M10 に, 位置変数を P10 に受け取ります。

■関連機能

機能	説明箇所
割込み	117 ページの「(3) 割込み」
通信	122 ページの「4.1.5 通信」
無条件分岐	114 ページの「(1) 無条件分岐・条件分岐・待機」

(5) タイマ

指定時間だけ待ったり、出力信号を指定時間幅でパルス出力することができます。

■命令語

命令語	説 明
Dly	指定時間のタイマとして機能します。

■文例

文例	説 明
Dly 0.05	’0.05 秒間だけ待ちます。
M_Out(10)=1 Dly 0.5	’出力信号ビット 10 を 0.5 秒間だけ ON します。

■関連機能

機能	説明箇所
パルス信号出力	121 ページの「(2) 信号出力」

(6) 停止

プログラムの実行を停止させることができます。移動中のロボットは減速停止します。

■命令語

命令語	説 明
Hlt	ロボットを停止し、プログラムの実行を中断します。始動により次行から実行します。
End	プログラムの 1 サイクルの終了を定義します。、連続運転時は End 命令実行にて再度先頭行より始動します。サイクル運転時、サイクル停止時は End 命令でプログラムが終了します。

■文例

文例	説 明
Hlt	’プログラムの実行を中断します。
If M_In(20)=1 Then Hlt	’入力信号ビット 20 が ON していたらプログラムを中断します。
Mov P1 WthIf M_In(18)=1, Hlt	’P1 へ移動中に入力信号ビット 18 が ON したらプログラムを中断します。
End	’プログラムの途中にあってもプログラムを終了します。

■関連機能

機能	説明箇所
付随節	280 ページの「Wth（ウィズ）」

4.1.4 外部信号の入出力

ロボットを外部機器（シーケンサなど）より制御する場合にもっとも一般的な信号制御の方法に関して説明します。

(1) 信号入力

シーケンサなどの外部機器から信号を取り込むことができます。
ロボット（システム）状態変数（M_In() など）を使って入力信号を確認します。ロボット（システム）状態変数についての詳細は 288 ページの「4.14 ロボット（システム）状態変数の詳細解説」を参照願います。

■命令語

命令語	説 明
Wait	入力信号が指定状態になるまで待機します。

■システム状態変数（ロボット状態変数）
M_In， M_Inb， M_Inw， M_DIn

■文例

文例	説 明
Wait M_In(1)=1	’入力信号ビット 1 が ON するまで待機します。
M1=M_Inb(20)	’数値変数 M1 に入力信号ビット 20 から 27 までの 8 ビットの状態を数値にして代入します。
M1=M_Inw(5)	’数値変数 M1 に入力信号ビット 5 から 20 までの 16 ビットの状態を数値にして代入します。

■関連機能

機能	説明箇所
信号出力	121 ページの「(2) 信号出力」
入力信号による分岐	114 ページの「(1) 無条件分岐・条件分岐・待機」
入力信号による割込み	117 ページの「(3) 割込み」

(2) 信号出力

シーケンサなどの外部機器へ信号を出力することができます。
ロボット（システム）状態変数（M_Out() など）を使って信号を出力します。ロボット（システム）状態変数についての詳細は 153 ページの「4.6 ロボット（システム）状態変数」を参照願います。

■命令語

命令語	説 明
Clr	汎用出力信号をパラメータの出力信号リセットパターンに基づいてクリアします。

■システム状態変数（ロボット状態変数）
M_Out， M_Outb， M_Outw， M_DOut

■文例

文例	説 明
Clr 1	’出力リセットパターンに基づいてクリアします。
M_Out(1)=1	’出力信号ビット 1 を ON します。
M_Outb(8)=0	’出力信号ビット 8 から 15 までの 8 ビットを OFF します。
M_Outw(20)=0	’出力信号ビット 20 から 35 までの 16 ビットを OFF します。
M_Out(1)=1 Dly 0.5	’出力信号ビット 1 を 0.5 秒間 ON します。（パルス出力）
M_Outb(10)=&H0F	’出力信号ビット 10 から 13 までの 4 ビットを ON に、14 から 17 までの 4 ビットを OFF します。

■関連機能

機能	説明箇所
信号入力	121 ページの「(1) 信号入力」
タイマ	119 ページの「(5) タイマ」

4.1.5 通信

パソコンなどの外部機器との間でデータ通信することができます。

■命令語

命令語	説 明
Open	通信回線をオープンします。
Close	通信回線をクローズします。
Print#	データを ASCII 形式で出力します。終了コードとして CR を出力します。
Input#	データを ASCII 形式で入力します。終了コードとして CR を出力します。
On Com GoSub	通信回線から割込みが発生した時に呼び出すサブルーチンを定義します。割込みは外部機器からデータが入力されたときに発生します。
Com On	通信回線からの割込み処理を許可します。
Com Off	通信回線からの割込み処理を禁止します。割り込みがあっても無効となります。
Com Stop	通信回線からの割込み処理を停止します。割込みがあった場合それを記憶し、許可された後で実行します。

■文例

文例	説 明
Open "COM1 : " As #1	' 通信回線 COM1 をファイル番号 1 としてオープンします。
Close #1	' ファイル番号 1 をクローズします。
Close	' オープンしているすべてのファイルをクローズします。
Print# 1, "TEST"	' ファイル番号 1 へ文字列 "TEST" を出力します。
Print# 2, "M1=" ; M1	' ファイル番号 2 へ文字列 "M1=" に続いて M1 の値を出力します。 出力データ例 : "M1=1"+CR(M1 の値が 1 の場合)
PRINT# 3, P1	' ファイル番号 3 へ位置変数 P1 の座標値を出力します。 出力データ例 : "(123.7, 238.9, 33.1, 19.3, 0, 0)(1, 0)"+CR (X=123.7、Y=238.9、Z=33.1、A=19.3、B=0、C=0、FL1=1、FL2=0 の場合)
Print# 1, M5, P5	' ファイル番号 1 へ数値変数 M5 の値と、位置変数 P5 の座標値を出力します。 M5 と P5 はカンマ (16 進 2C) で区切られます。 出力データ例 : "8, (123.7, 238.9, 33.1, 19.3, 0, 0)(1, 0)"+CR (M5=8、P5 は X=123.7、Y=238.9、Z=33.1、A=19.3、B=0、C=0、FL1=1、FL2=0 の場合)
Input# 1, M3	' 入力したデータを数値にして数値変数 M3 に代入します。 入力データ例 : "8"+CR(値 8 を代入する場合)
Input# 1, P10	' 入力したデータを座標値にして位置変数 P10 に代入します。 入力データ例 : "(123.7, 238.9, 33.1, 19.3, 0, 0)(1, 0)"+CR (P5 は X=123.7、Y=238.9、Z=33.1、A=19.3、B=0、C=0、FL1=1、FL2=0 になります)
Input# 1, M8, P6	' 入力した最初のデータを数値にして数値変数 M8 に, カンマに続く次のデータを座標値にして位置変数 P6 に代入します。M8 と P6 はカンマ (16 進 2C) で区切ります。 入力データ例 : "7, (123.7, 238.9, 33.1, 19.3, 0, 0)(1, 0)"+CR (M8=7、P6 は X=123.7、Y=238.9、Z=33.1、A=19.3、B=0、C=0、FL1=1、FL2=0 になります)
On Com(2) GoSub *RECV	' 通信回線 COM2 へデータが入力されたらラベル RECV の行のサブルーチンを呼び出すことを定義します。
Com(1) On	' 通信回線 COM1 からの割込みを許可します。
Com(2) Off	' 通信回線 COM2 からの割込みを禁止 (無効) します。
Com(1) Stop	' 通信回線 COM1 からの割込みを停止 (保留) します。

■関連機能

機能	説明箇所
サブルーチン	118 ページの「(4) サブルーチン」
割込み	117 ページの「(3) 割込み」

4.1.6 式と演算

以下に、使用できる演算子とその意味、および文例を示します。

(1) 演算子一覧

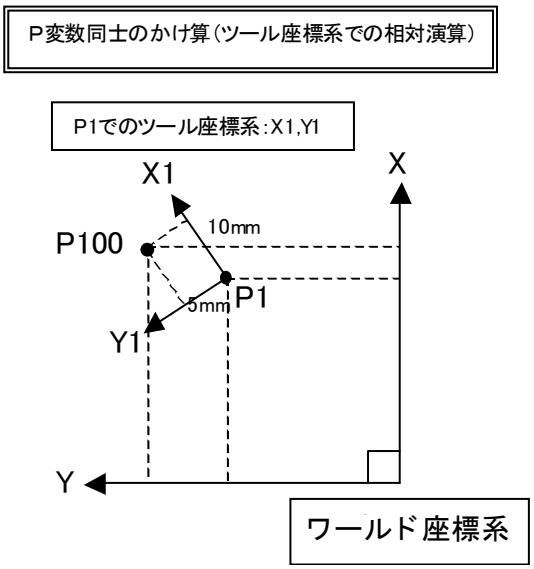
区分	演算子	意味	文例	
代入	=	右辺を左辺に代入	P1=P2	; 位置変数 P1 に P2 を代入します。
			P5=P_Curr	; 現位置変数 P5 に現在の座標値を代入します。
			P10.Z=100.0	; 位置変数 P10 の Z 座標値を 100.0 にします。
			M1=1	; 数値変数 M1 に値 1 を代入します。
			STS$="OK"	; 文字列変数 STS$ に OK という文字列を代入します。
数値演算	+	加算	P10=P1+P2 注 1)	; 位置変数 P10 に P1 と P2 の各座標成分同士を加算した結果を代入します。
			Mov P8+P9	; 位置変数 P8 と P9 の各座標成分同士を加算した位置へ移動します。
			M1=M1+1	; 数値変数 M1 の値に 1 を加えます。
			STS$="ERR"+"001"	; 文字列変数 STS$ に ERR という文字列と 001 という文字列をつなぎ合わせて代入します。
	-	減算	P10=P1-P2	; 位置変数 P10 に P1 から P2 の各座標成分を減算した結果を代入します。
			Mov P8-P9	; 位置変数 P8 から P9 の各座標成分同士を減算した位置へ移動します。
			M1=M1-1	; 数値変数 M1 の値から 1 を引きます。
	*	乗算	P1=P10*P3 注 2)	; 位置変数 P1 へ P10 から P3 への相対変換結果を代入します。
			M1=M1*5	; 数値変数 M1 の値を 5 倍します。
	/	除算	P1=P10/P3	; 位置変数 P1 へ P10 から P3 への逆相対変換結果を代入します。
			M1=M1/2	; 数値変数 M1 の値を 2 で割ります。
	^	指数演算	M1=M1 ^ 2	; 数値変数 M1 の値を二乗します。
	¥	整数除算	M1=M1¥3	; 数値変数 M1 の値を 3 で割って整数にします。(切り捨て)
	Mod	剰余演算	M1=M1 Mod 3	; 数値変数 M1 の値を 3 で割った余りにします。
	-	符号反転	P1=-P1	; 位置変数 P1 の各座標成分の符号を反転します。
			M1=-M1	; 数値変数 M1 の値の符号を反転します。
比較演算	=	等しいかどうか比較	If M1=1 Then *L2	; 数値変数 M1 の値が 1 ならラベル L2 へ分岐します。
			If STS$="OK" Then *L1	; 文字列変数 STS$ が OK という文字列ならばラベル L1 へ分岐する。
	<> または ><	等しくないかどうかを比較	If M1<>2 Then *L3	; 数値変数 M1 の値が 2 でないならラベル L3 へ分岐します。
			If STS$<>"OK" Then *L9	; 文字列変数 STS$ が OK という文字列でないならばラベル L9 へ分岐します。
	<	より小さいかどうかを比較	If M1<10 Then *L3	; 数値変数 M1 の値が 10 より小さいならラベル L3 へ分岐します。
			IF Len(STS$)<3 Then *L1	; 文字列変数 STS$ の文字数が 3 より小さいならラベル L1 へ分岐します。
	>	より大きいかどうかを比較	If M1>9 Then *L2	; 数値変数 M1 の値が 9 より大きいならラベル L2 へ分岐します。
			If Len(STS$)>2 Then *L3	; 文字列変数 STS$ の文字数が 2 より大きいならラベル L3 へ分岐します。
	=< または <=	等しいかより小さいかどうかを比較	If M1<=10 Then *L2	; 数値変数 M1 の値が 10 か 10 より小さいならラベル L2 へ分岐します。
			If Len(STS$)<=5 Then *L3	; 文字列変数 STS$ の文字数が 5 か 5 より小さいならラベル L3 へ分岐します。
	=> または >=	等しいかより大きいかどうかを比較	If M1>=11 Then *L2	; 数値変数 M1 の値が 11 か 11 より大きいならラベル L2 へ分岐します。
			If Len(STS$)>=6 Then *L3	; 文字列変数 STS$ の文字数が 6 か 6 より大きいならラベル L3 へ分岐します。
論理演算	And	論理積演算	M1=M_Inb(1) And &H0F	; 数値変数 M1 に入力信号ビット 1 から 4 までの状態を数値にして代入します。(入力信号ビット 5 から 8 の状態は OFF とします。)
	Or	論理和演算	M_Outb(20)=M1 Or &H80	; 出力信号ビット 20 から 27 に数値変数 M1 の値を出力します。この時、出力信号ビット 27 は常に ON となります。
	Not	否定演算	M1=Not M_Inw(1)	; 数値変数 M1 に入力信号ビット 1 から 16 までの状態を反転し、数値にして代入します。
	Xor	排他的論理和演算	M2=M1 Xor M_Inw(1)	; 数値変数 M2 に、M1 と入力信号ビット 1 から 16 の状態の排他的論理和をとり、数値にして代入します。
	<<	論理左シフト演算	M1=M1 << 2	; 数値変数 M1 を 2 ビット左にシフトします。
	>>	論理右シフト演算	M1=M1 >> 1	; 数値変数 M1 を 1 ビット右にシフトします。

注 1) 124 ページの「位置データの相対演算 (加算)」を参照してください。

注 2) 124 ページの「位置データの相対演算 (乗算)」を参照してください。

(2) 位置データの相対演算 (乗算)

数値変数の演算は通常の四則演算ですが、位置変数の演算は単純な四則演算ではなく座標変換をおこないます。以下に簡単な例を元に説明します。

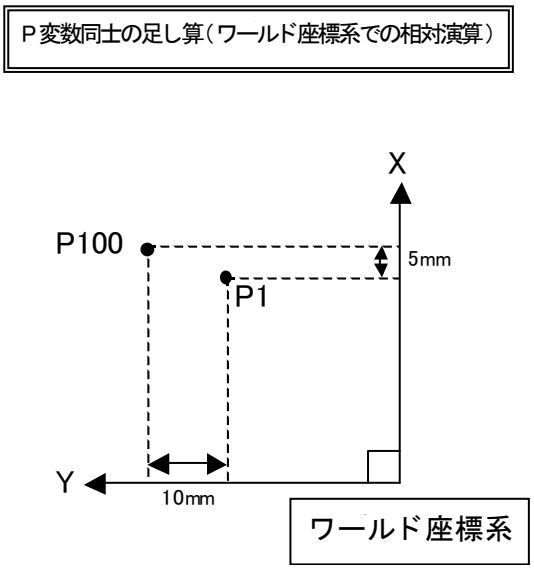


相対演算の例 (乗算)

```
1 P2=(10,5,0,0,0,0)(0,0)
2 P100=P1*P2
3 Mov P1
4 Mvs P100
P1=(200,150,100,0,0,45)(4,0)
```

この例はティーチ位置 P1 にて P1 のツール座標系内で相対移動させた場合の例です。
P2 の X, Y の値がツール座標系内での移動量になります。
P 変数同士のかけ算が相対演算になります。
かけ算の順番により意味が異なりますのでご注意ください。
相対移動量を設定した変数 (P2) は後ろに記述します。
P2 の姿勢軸部分 (A,B,C) がゼロの場合は P1 の姿勢をそのまま使用します。値がある場合は P1 の姿勢を元に Z, Y, X 回転 (C, B, A 順) に回転させた姿勢になります。乗算はツール座標系内での加算ですが、除算の場合はツール座表系内での減算になります。

(3) 位置データの相対演算 (加算)



相対演算の例 (加算)

```
1 P2=(5,10,0,0,0,0)(0,0)
2 P100=P1+P2
3 Mov P1
4 Mvs P100
P1=(200,150,100,0,0,45)(4,0)
```

この例はティーチ位置 P1 にてワールド座標系内で相対移動させた場合の例です。
P2 の X, Y の値がワールド座標系内での移動量になります。P 変数同士のたし算が相対演算になります。
P2 の C 軸部分に値を入れると P100 の C 軸の回転成分も変更できます。P1 の C 軸成分に P2 の C 軸成分を足し算した値になります。

注意) 上記例では簡単に説明するため二次元で解説していますが、実際は三次元で演算します。
また、ツール座標系は各ロボットの姿勢によって異なります。

4.1.7 付随節

移動命令に処理を付加できます。

■付随節

付随節	説 明
Wth	移動命令に無条件で処理を付加します。
WthIf	移動命令に条件付で処理を付加します。

■文例

文例	説 明
Mov P1 Wth M_Out(20)=1	’P1 への移動開始と同時に出力信号ビット 20 を ON します。
Mov P1 WthIf M_In(20)=1, Hlt	’P1 へ移動中に入力信号ビット 20 が ON したら停止します。
Mov P1 WthIf M_In(19)=1, Skip	’P1 へ移動中に入力信号ビット 19 が ON したら P1 への移動を中断して次の行へ進みます。

■関連機能

機能	説明箇所
関節補間動作	100 ページの「(1) 関節補間動作」
直線補間動作	101 ページの「(2) 直線補間動作」
円弧補間動作	102 ページの「(3) 円弧補間動作」
停止	120 ページの「(6) 停止」

4.2 MELFA-BASIC VとMELFA-BASIC IVの違い

4.2.1 MELFA-BASIC Vについて

CR7xx シリーズコントローラでは、ロボット言語に MELFA-BASIC V を搭載しました。この MELFA-BASIC V は、従来の CRn-500 シリーズコントローラに搭載されていた MELFA-BASIC IV をベースに、より使いやすく、より記述しやすくしたものです。

ここでは、その違いを解説します。

4.2.2 MELFA-BASIC Vの特長

MELFA-BASIC V は、MELFA-BASIC IV と比較して、次のような特長を持っています。

(1) 行番号を不要としました。従来の行番号はステップ番号として、自動的に採番します。
これにより、行番号をつける煩わしさがなくなり、プログラミングの効率の向上や、デバッグ時における行番号誤りによるロスの低減が図れます。

(2) 命令や変数名などに小文字を使用可能としました。これにより、可読性が大幅に向上します。

(3) 命令や関数を追加し、機能向上を図りました。

4.2.3 MELFA-BASIC IVとの比較

MELFA-BASIC V と IV の比較を、表 4-2 に示します

表 4-2 : MELFA-BASIC IV との比較

項目	MELFA-BASIC V	MELFA-BASIC IV
プログラム名	12 文字以内の英大文字と数字 (O/P 表示のため 4 文字以内を推奨)	
使用可能文字	・英字 (大文字、小文字) ・数字 ・記号	・英字 (大文字のみ。小文字はコメントと文字列データにのみ使用可能) ・数字 ・記号
ステップ番号 (行番号)	ステップ番号として、プログラム登録時に自動採番します。	行番号として、プログラミング時に入力する必要があります。
1 行の長さ	240 文字以内	127 文字以内
変数名	16 文字以内。変数名に英大文字と英小文字が使用可能。大文字と小文字の区別はしませんが、読み出し時に最初に登録された表記に変換します。	8 文字以内。変数名に使用される英字はすべて大文字に変換されます。
ラベル名	16 文字以内。ラベル名に英大文字と英小文字が使用可能。大文字と小文字の区別はしませんが、読み出し時に最初に登録された表記に変換します。	8 文字以内。ラベル名に使用される英字はすべて大文字に変換されます。
命令語 関数 システム状態変数	英大文字と英小文字を組み合わせて定義しています。登録時に、大文字と小文字の区別はしませんが、読み出し時にシステム登録された表記に変換します。	すべて大文字で定義しています。登録時も大文字で登録します。
分岐命令 (GoTo、GoSub) の飛び先指定の方法	ラベルで指定します。	ラベルまたは行番号で指定します。

4.3 マルチタスクの機能

4.3.1 マルチタスクとは

ここでは、マルチタスクの機能を解説します。
マルチタスクとは複数のプログラムを並列して実行させる機能で、ロボットプログラムでの周辺機器の制御を可能にする機能です。
マルチタスクの実行は、タスクスロットと呼ばれる入れ物(全部で 32 本あります、出荷時設定は 8 本まで)に並列実行するプログラムを入れておこないます。

マルチタスクの実行の開始は、操作パネルからの始動や専用入力信号からの始動、またはマルチタスクに関する命令を実行しておこないます。

図 4-1 にマルチタスクの実行環境を示します。

マルチタスクのスロット環境

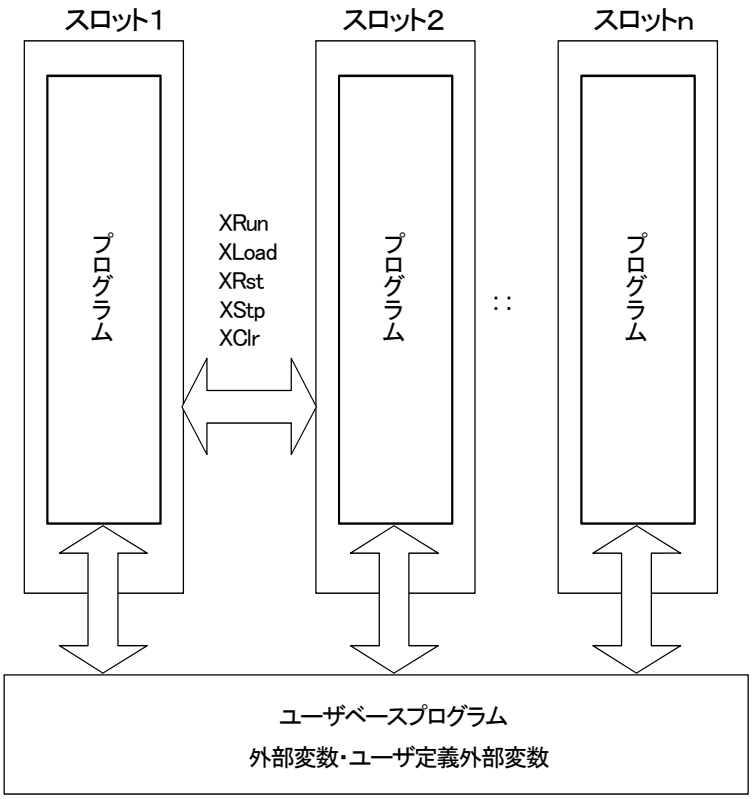


図 4-1：マルチタスクの実行環境

◇◆◇プログラムの実行について◇◆◇
プログラムの実行は、〝タスクスロット〟と呼ばれる入れ物に実行するプログラムを入れておこなっています。
例えば、一つのプログラムを運転する場合(通常、コントローラの操作パネルでプログラムを選択して始動する場合)ではコントローラシステムが、操作パネルで選択されたプログラムを暗黙のうちにタスクスロット 1 に入れて実行しています。

4.3.2 マルチタスクの実行方法

マルチタスクの実行方法は表 4-3 に示す 3 種類があります。

表 4-3：マルチタスクの実行方法

	実行種類	解説
1	プログラムからの実行	プログラム中の任意の位置から MELFA-BASIC V 命令で、プログラムの並列実行を開始する方法です。並列実行するプログラムの指定や並列実行中のプログラムの停止も可能です。 この方法は、プログラムの流れにより、並列実行するプログラムを選択する場合に有効です。 関連する命令には、XLoad（エックスロード）、XRun（エックスラン）、XStp（エックスストップ）、XRst（エックスリセット）、XClr（エックスクリア）があります。詳細は、本書 /164 ページの「4.13 命令の詳細説明」の各頁を参照してください。
2	コントローラの操作パネル、または外部入出力信号からの実行	パラメータ「SLT*」の設定情報（起動条件が "START"、もしくは "ERROR"）に基づき、始動操作による並列実行の開始や常時並列実行、またはエラー発生時並列実行を開始する方法です。予めパラメータ「SLT*」を設定しておく必要があります。この方法は、プログラムの流れに依存せず、予め決まった形での同時実行、あるいは順次実行をおこなう場合に有効です。
3	電源 ON にて自動的に実行	コントローラの電源を ON した直後に常時実行させることが可能です。 SLT* パラメータの起動条件に "ALWAYS" と指定したプログラムはコントローラの電源が ON した直後から常時実行されます。 これにより入出力信号の監視用のタスクスロットなどはシーケンサ側からの起動をする必要がなくなります。 さらに動作用のプログラムを常時実行しているプログラムから実行させることも可能です。この場合はパラメータの「ALWENA」の値を 1 にすることで常時実行プログラム内から XRun、XLoad などの X** 命令や Servo 命令、Reset 命令を実行することができます。

4.3.3 各タスクスロットの運転状態

各タスクスロットの運転状態は、操作やコマンドにより、図 4-2 のように状態が変化します。
それぞれの状態は、ロボット（システム）状態変数や、外部出力信号で確認することができます。

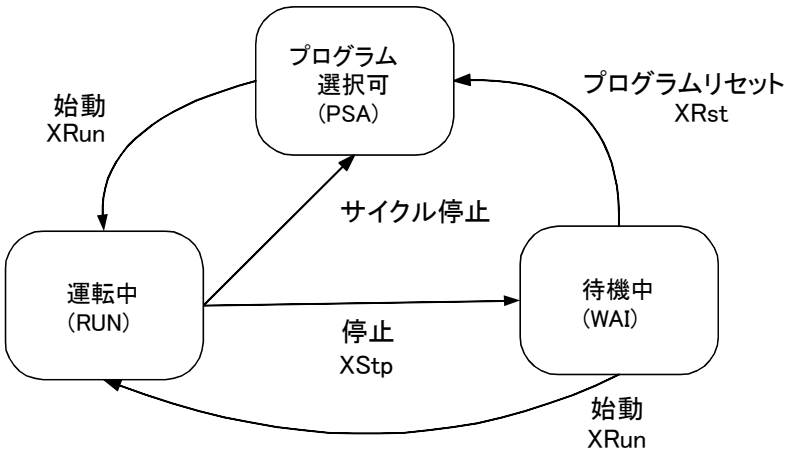


図 4-2：各タスクスロットの運転状態

＜タスクスロットに関するパラメータについて＞

32 本（出荷時 8 本）ある各タスクスロット毎に、使用するプログラム名、運転形態、起動条件、優先度を設定するもので SLT1 ～ SLT32 があります。
パラメータの詳細は 408 ページの「5 パラメータでの設定機能」を参照願います。

■指定形式

SLT* = ①プログラム名、②運転形態、③起動条件、④優先度

■各設定値とその意味

パラメータの項目	初期値	設定値	解説
①プログラム名	SLT1：操作パネルにて選択したプログラム SLT2 ～ 32：パラメータにて設定したいプログラム名	登録済みのプログラムを設定可能	マルチタスク運転にて決まったプログラムを実行する場合にパラメータにて設定します。 実行するプログラムが状況により可変になる場合はプログラムにて XLoad、XRun 命令などにより指定することも可能です。 SLT1 は操作パネルにて選んだプログラムが設定されます。
②運転形態	REP	・REP：連続運転	REP に指定した場合はプログラムの最終行または End 命令実行にて再度先頭から実行されます。
		・CYC：1 サイクル運転	CYC に指定した場合は 1 サイクル運転後停止し、選択状態は解除されます。プログラム選択状態を残したい場合はパラメータの SLOTON を変更してください。詳細は 408 ページの「5 パラメータでの設定機能」の SLOTON の項を参照願います。
③起動条件	START	・START：操作パネルの START ボタン、IO の START 信号による実行	START は通常に起動させたい場合に設定します。注 1)
		・ALWAYS：コントローラの電源 ON にて実行	ALWAYS は常時実行させたい場合に使用します。 ただし、Mov などの動作命令は実行できません。常時実行プログラムの停止は、XStp 命令にて可能です。操作パネルや外部入力信号からの停止、および非常停止では停止しません。
		・ERROR：コントローラがエラー状態になった時に実行	ERROR はエラー発生時に実行したい場合に指定します。 Mov 命令などロボットを動作させる命令の実行はできません。運転形態（REP/CYC）はその設定値にかかわらず 1 サイクル運転（CYC）になります。
④優先度 （優先実行行数）	1	1 ～ 31：マルチタスク運転時に一度に実行する行数	数字を大きくするとそのタスクスロットの一度に実行する行数が増えます。 例えば SLT1 は 10、SLT2 は 5、SLT3 は 1 と指定した場合は SLT1 のプログラムを 10 行実行したあと SLT2 のプログラムを 5 行実行、SLT3 のプログラムを 1 行実行します、あとはその繰り返しになります。

注 1）操作パネルからの始動操作、および専用入力信号の START 信号からの始動は、起動条件が "START" に設定されている全てのタスクスロットのプログラムが同時に実行を開始します。
個別に始動する場合は、専用入力信号（S1START ～ S32START）から各タスクスロット単位で始動します。この場合、同専用入出力パラメータに予め信号番号を割り付けて使用します。専用入出力の割り付けについての詳細は、本書 /509 ページの「6.2 シーケンサリンク入出力機能」を参照願います。

■設定例

タスクスロット 2 に以下を指定する場合のパラメータ設定例を示します。

指定内容）プログラム名：5
運転形態：　連続運転
起動条件：　常時実行
優先度：　10
SLT2=5, REP, ALWAYS, 10

4.3.4 マルチタスクプログラム作成上の注意点

(1) タスク本数と処理時間の関係

マルチタスク時は複数のロボットプログラムを並列に処理しているように見せますが実際はプログラムを切り替えながら 1 行ごと (行数は変更可能　408 ページの「5 パラメータでの設定機能」「SLTn」パラメータ参照) に処理しています。よってタスク本数が増えると全体のプログラム実行時間が延びることになります。マルチタスクを使用する場合、タスク本数は必要最小限にしてください。

ただし、動作命令 (Mov や Mvs 命令など) 実行中の他のタスクのプログラムは随時処理します。

(2) 並列実行するプログラム本数の最大値指定

並列実行するプログラムの本数はパラメータ「TASKMAX」パラメータ (408 ページの「5 パラメータでの設定機能」を参照してください) で設定します。(初期値は 8)9 本以上のプログラムを並列実行する場合、このパラメータを変更してください。

(3) 外部変数でのプログラム間のデータの受け渡し方法

マルチタスク運転しているプログラム間のデータの受け渡しは M_00 や P_00 などのプログラム外部変数 (145 ページの「4.4.23 プログラム外部変数」参照) やユーザ定義外部変数 (146 ページの「4.4.24 ユーザ定義外部変数」参照) を使用しておこないます。

以下に例を示します。この例では、入力信号 8 のオン / オフ状態をタスクスロット 2 のプログラムで判断し、オンしたことを外部変数の M_00 を使ってタスクスロット 1 のプログラムに伝えるものです。

```
< タスクスロット 1>
   1 M_00=0                              ; M_00 に 0 を代入する。
   2 *L2 :    If M_00=0 Then *L2         ; M_00 の値が 0 でなくなるまで待つ。
   3 M_00=0                              ; M_00 に 0 を代入する。
   4 Mov P1                              ; 以下目的作業を進める。
   5 Mov P2
          :
   100 GoTo *L2                          ; STEP 番号 2 から繰り返す。
```

```
< タスクスロット 2>( 信号や変数のプログラム )
   1 If M_In(8)<>1 Then *L4              ; 入力信号 8 がオンしていなかったらラベル L4 へ分岐
                                           する。
   2 M_00=1                              ; M_00 に 1 を代入する。
   3 M_01=2                              ; 以下目的作業を進める。
   4 *L4
       :
```

(4) ロボット (システム) 状態変数でのプログラムの運転状態の確認

マルチタスク運転してるプログラムの状態はロボット (システム) 状態変数 (M_Run、M_Wai、M_Err) を使用して、どのタスクスロットからも参照することができます。

例)　M1=M_Run(2)　タスクスロット 2 の運転中状態を得る。

ロボット (システム) 状態変数についての詳細は 154 ページの「4.6 ロボット (システム) 状態変数」を参照願います。

(5) ロボット動作のプログラムは基本的にタスクスロット 1 で実行

Mov 命令など、ロボット本体の動作を記述したプログラムは基本的にタスクスロット 1 に設定して実行します。タスクスロット 1 以外で動作させる場合には、ロボット本体の獲得と開放命令 (GetM、RelM) を使用する必要があります。命令の詳細は、本書 /164 ページの「4.13 命令の詳細説明」の各頁を参照してください。

(6) 常時実行プログラムでの初期化処理

起動条件を ALWAYS に設定したタスクスロットのプログラムは、その運転形態を REP に指定すると連続運転 (くり返し運転) されます。このプログラムで初期化処理をおこなう場合は、初期化完了フラグを設けて条件分岐させるなど、初期化処理を 2 度以上通らないように作成してください。

(運転形態を CYC に指定 (1 サイクル運転) したタスクプログラムは 1 度しか実行されませんのでこのような工夫は不要です。)

◇◆◇タスクスロット 1 にはメカ 1 が割り当てられています◇◆◇
初期状態では、タスクスロット 1 にメカ 1(標準システムのロボット本体) が自動的に割り当てられています。
このため、タスクスロット 1 ではメカを獲得しなくても (GetM 命令を実行しなくても) 移動命令を実行できますが、タスクスロット 1 以外で移動命令を実行する場合には、タスクスロット 1 のメカ獲得状態を解除して (RelM 命令を実行して)、移動命令を実行するタスクスロットでメカを獲得してください。(GetM 命令を実行してください。)

4.3.5 マルチタスクプログラム運転上の注意点

(1) マルチタスクの開始

操作パネルからの始動操作、および専用入力信号の START からの始動は、タスクスロットパラメータの起動条件に "START" と設定されている全てのタスクスロットのプログラムが同時に実行を開始します。

専用入力信号の S1START ～ S32START からの始動は、各タスクスロット毎に実行を開始することができます。この場合、同専用入出力パラメータに予め信号番号を割り付けて使用します。専用入出力の割り付けについての詳細は、本書 /509 ページの「6.2 シーケンサリンク入出力機能」を参照願います。

(2) 運転状態の表示

操作パネルの [START]、[STOP] スイッチの LED、および専用入出力信号の START、STOP は、パラメータ「SLT*」の起動条件に "START" と設定してあるタスクスロットのプログラムの運転状態を表示しています。1 つでもプログラムが運転していれば [START] スイッチの LED が点灯し、専用出力信号の START がオンします。また全てのプログラムが停止すれば [STOP] スイッチの LED が点灯し、専用出力信号の STOP がオンします。

専用出力信号の S1START ～ S32START、および S1STOP ～ S32STOP は、タスクスロット毎の個別の運転状態を出力します。個別の運転状態を知る必要がある場合は、同専用入出力パラメータに信号番号を割付けて外部信号で確認してください。専用入出力の割り付けについての詳細は、本書 /509 ページの「6.2 シーケンサリンク入出力機能」を参照願います。

起動条件が ALWAYS、ERROR に設定されているプログラムの状態は [START]、[STOP] スイッチの LED には影響しません。常時実行プログラムの運転状況を確認するにはパソコンサポートソフトウェア (オプション) のモニタリングツールで確認できます。

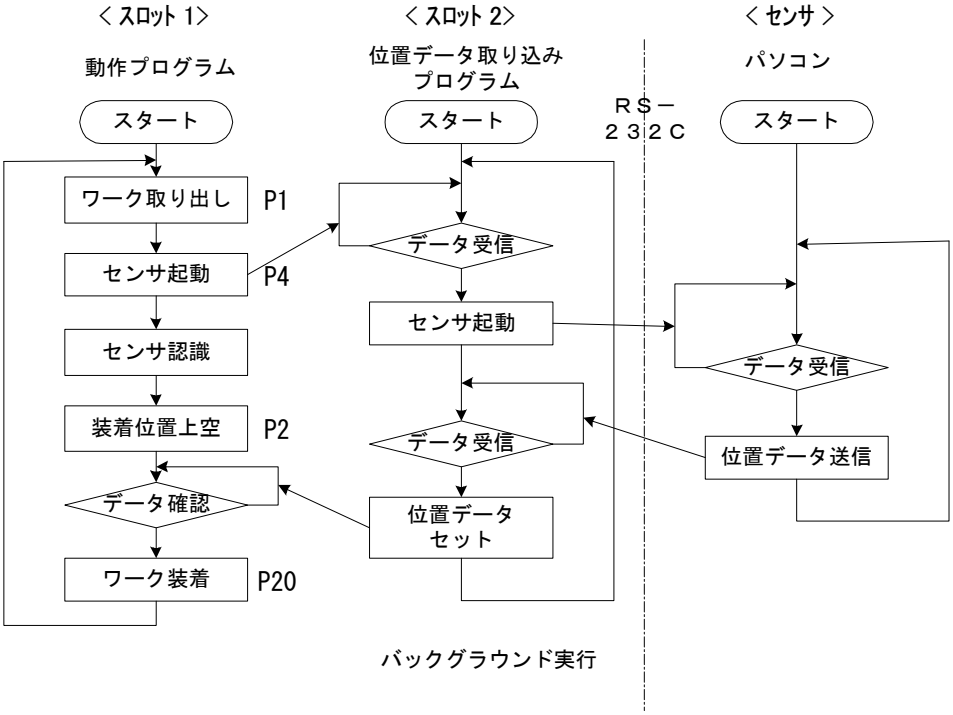
4.3.6 マルチタスクの使用例

ここでは、マルチタスクの実行例を紹介します。

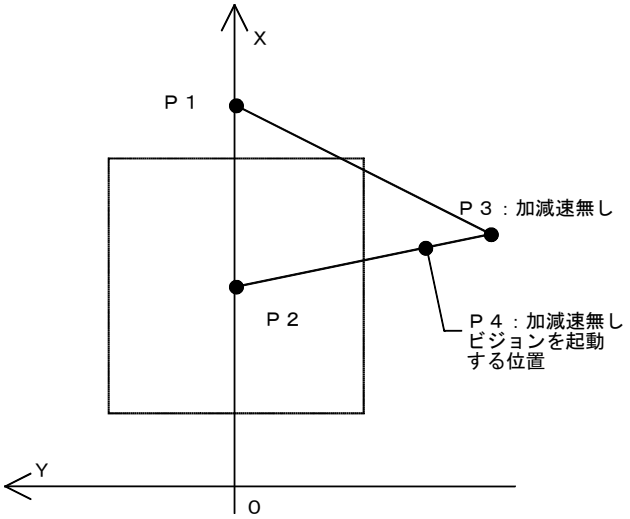
(1) ロボットの作業内容

ロボットプログラムは「動作プログラム」と「位置データ取り込みプログラム」の 2 本です。「動作プログラム」はタスクスロット 1 で、「位置データ取り込みプログラム」はタスクスロット 2 で実行することとします。ロボットが動作途中で、センサへの起動指令を出力すると、位置データ取り込みプログラムを介してパソコンへデータ要求がなされます。パソコンはデータ要求にもとづき、位置データをロボットへ送信します。ロボット側は位置データ取り込みプログラムを介して、補正データを取り込みます。

＜処理フロー＞



P1：ワーク取り位置（吸着タイマ Dly 0.05）
P2：ワーク置き位置（解除タイマ Dly 0.05）
P3：ビジョン前位置（通過点 Cnt で止まらず）
P4：ビジョンシャッター位置（通過点 Cnt で止まらず）
P_01：ビジョン補正データ
P20：ビジョン補正データを P2 に加えた位置（相対演算）



(2) マルチタスク実行までの手順

■手順 1：プログラムの作成

<1> 動作プログラム (プログラム名 : 1)

```
1 Cnt 1                                  ' 軌跡接続動作を有効
2 Mov P2,10                              'P2 上空 +10mm へ移動
3 Mov P1,10                              'P1 上空 +10mm へ移動
4 Mov P1                                 'P1 ワーク取り位置へ
5 M_Out(10)=0                            ' ワーク取り
6 Dly 0.05                               ' タイマー 0.05 秒
7 Mov P1,10                              'P1 上空 +10mm へ移動
8 Mov P3                                 ' ビジョン前位置 P3 へ移動
9 Spd 500                                ' 直線速度 500mm/ 秒に設定
10 Mvs P4                                'P4 通過でビジョン取り込み起動
11 M_02#=0                               ' インターロック変数 (M_01=1/M_02=0) で
12 M_01#=1                               ' バックグラウンド処理でデータ取り込み開始
13 Mvs P2,10                             'P2 上空 +10mm へ移動
14 *L2: If M_02#=0 Then GoTo *L2         ' インターロック変数 M_02=1 になるまで待ち
15 P20=P2*P_01                           'P20 へビジョン補正 P_01 を加え上空 +10mm へ移動
16 Mov P20,10                            'P20 上空 +10mm へ移動
17 Mov P20                               'P20 ワーク置き位置へ
18 M_Out(10)=1                           ' ワーク置き
19 Dly 0.05                              ' タイマー 0.05 秒
20 Mov P20,10                            'P20 上空 +10mm へ移動
21 Cnt 0                                 ' 軌跡接続移動を無効
22 End                                   '1 サイクル終了
```

<2> 位置データ取り込みプログラム (プログラム名 : 2)

```
1 *L1: If M_01#=0 Then GoTo *L1          ' インターロック変数 M_01=1 になるまで待ち
2 Open "COM1:" As #1                     'EThernet 回線オープン
3 Dly M_03#                              ' 仮想処理タイマー (0.05 秒 )
4 Print #1,"SENS"                        ' 文字列 "SENS" を EThernet( ビジョン側 ) へ送信
5 Input #1,M1,M2,M3                      ' ビジョン補正値 ( 相対データ ) の取り込み待ち
6 P_01.X=M1                              'ΔX 座標の代入
7 P_01.Y=M2                              'ΔY 座標の代入
8 P_01.Z=0.0                             '
9 P_01.A=0.0                             '
10 P_01.B=0.0                            '
11 P_01.C=Rad(M3)                        'ΔC 座標の代入
12 Close                                 'EThernet 回線クローズ
13 M_01#=0                               ' インターロック変数 M_01=0
14 M_02#=1                               ' インターロック変数 M_02=1
15 End                                   ' 処理終了
```

■手順 2：タスクスロットパラメータの設定

以下のようにスロットパラメータを設定します。

パラメータ	プログラム名	運転モード	運転形態	実行行数
SLT1	1	REP	START	1
SLT2	2	REP	START	1

■手順 3：タスクスロットパラメータの反映

パラメータ (SLT1, SLT2) を有効にするために一度電源を OFF → ON します。

■手順 4：始動

操作パネルからの始動でプログラム 1 とプログラム 2 の運転が開始します。

4.3.7 プログラム容量について

ロボットプログラムを扱う領域には、保存、編集、実行の 3 種類があります。各領域の容量は、「表 4-4 プログラム各領域の容量」を参照ください。

(1) プログラム保存領域

プログラムを保存する領域です。標準では合計 920K バイトのプログラムが保存可能です。

(2) プログラム編集領域

プログラムの編集、ステップ実行で動作確認する際に使用する領域です。プログラム編集領域は 380K バイトで、これがプログラム 1 本あたりの最大値となります。

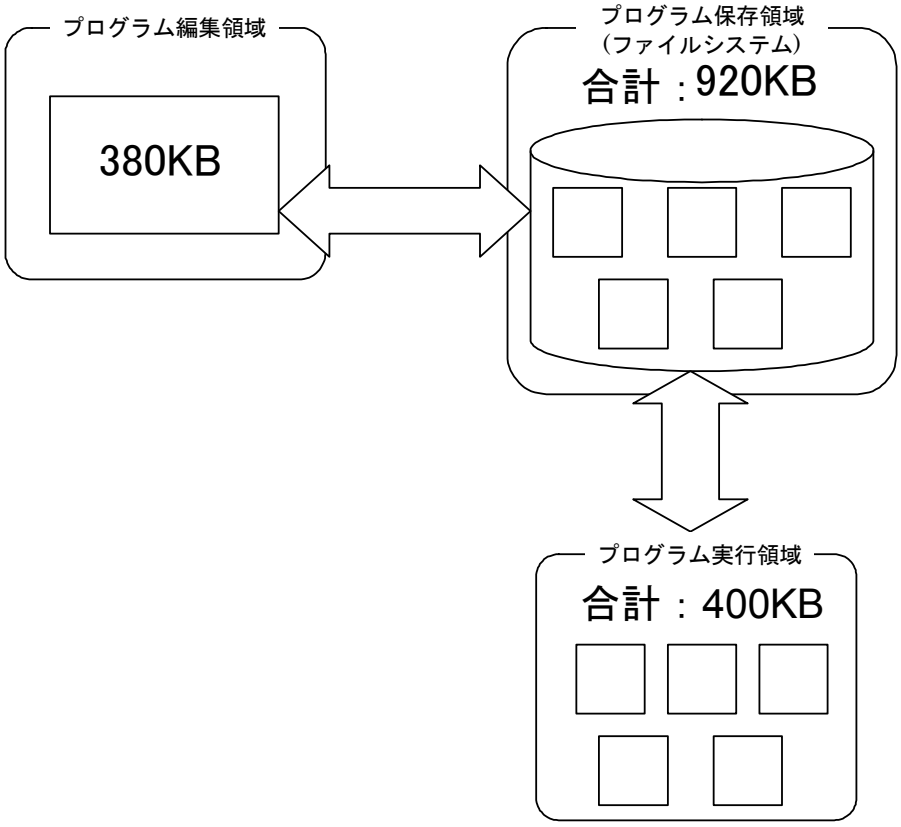
(3) プログラム実行領域

プログラム実行領域は、プログラムを自動運転する際に使用する領域です。プログラム実行領域は 400K バイトです。ユーザベースプログラム、マルチタスク、XRun、CallP 命令により同時に実行領域にロードされるプログラムの合計容量が 400K バイト以下である必要があります。

表 4-4：プログラム各領域の容量

名称	容量
	標準メモリ時
(1) プログラム保存領域	920K バイト
(2) プログラム編集領域	380K バイト
(3) プログラム実行領域	400K バイト

各プログラムの容量はティーチングボックスやパソコンサポートソフトウェア (RT ToolBox2) のプログラム管理画面で確認することができます。



4.4 MELFA-BASIC Vの詳細仕様

ここでは、MELFA-BASIC V の書式や構成などの文法についての詳細説明と、各命令語の機能の詳細を示します。以下に命令文を構成する要素を説明します。

(1) プログラム名

プログラム名は 12 文字以内で作成してください。ただし、操作パネルの表示が 4 文字までしか表示できないため **4 文字以内**で作成されることを推奨します。また使用可能な文字は以下です。

区分	使用可能な文字
英字(半角)	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z (半角大文字のみ使用してください。小文字、または全角文字で登録すると正常に実行できない場合があります)
数字(半角)	0 1 2 3 4 5 6 7 8 9

4 文字以上で作成した場合は操作パネルからの選択はできません。また、外部入力信号から選択させる場合は番号で作成してください。CallP 命令にてサブプログラムとして実行する場合は 4 文字以上で英文字の名前でもかまいません。ただし、操作パネルからは選択できません。

(2) 命令文

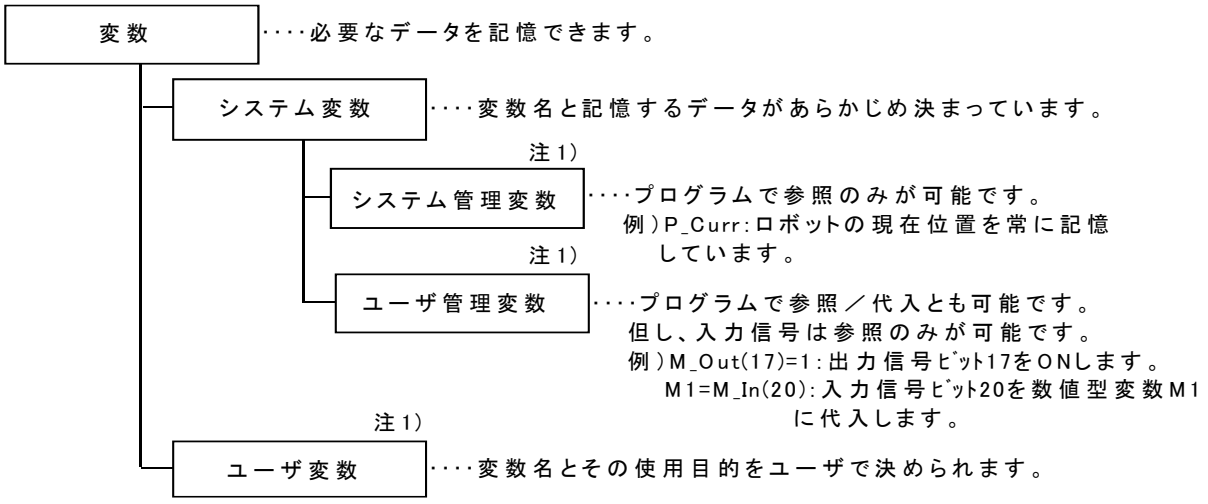
命令文の構成例。

1 Mov P1 Wth M Out(17)=1
① ② ③ ④

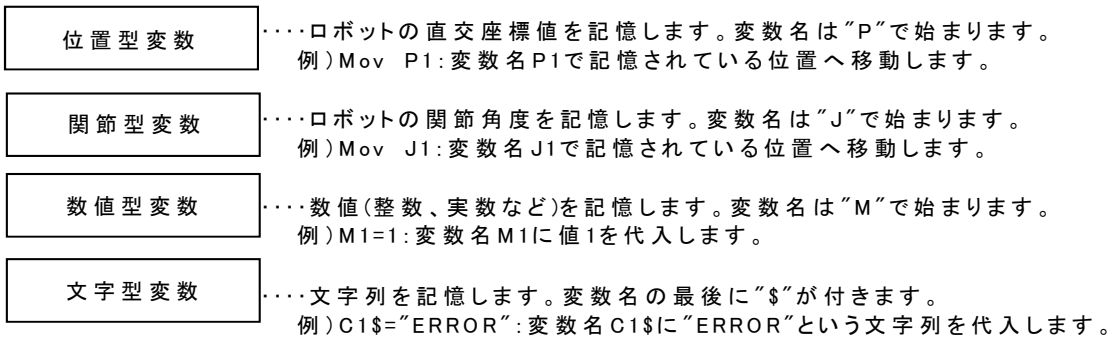
①ステップ番号：プログラムの実行順序を決めるための番号です。番号の昇順に実行します。
②命令語：ロボットの動作や作業を指定する命令です。
③データ：命令語毎に必要となる変数や数値などのデータです。
④付随節：必要に応じてロボットの作業を付加するときに指定します。

(3) 変数

プログラムで以下に示す種類の変数が使用できます。



注1) 各変数はそれぞれ以下の種類に分類されます。



4.4.1 文

文は、プログラムを構成する最小単位で、命令語とそれに与えるデータから成り立ちます。

例） Mov　　P1
　　命令語 データ
　　命令文

4.4.2 付随文

移動命令に限って、付随節によって命令文同士をつなぐことができます。
これにより移動命令と並行して一部のコマンドが実行できます。

例） Mov　P1　Wth　M_Out(17)=1
　　命令文　付随節　　命令文

詳細については 280 ページの「Wth（ウィズ）」または 280 ページの「WthIf（ウィズイフ）」及び各移動命令（Mov（ムーブ），Mva（ムーブアーチ），Mvs（ムーブエス），Mvr（ムーブアール），Mvr2（ムーブアール 2），Mvr3（ムーブアール 3））を参照ください。

4.4.3 行

行は、ステップ番号と 1 つの命令文から成り立ちます。ただし付随節を用いた場合、命令文は 2 つになります。一行の長さは、240 文字までです。（行末文字は含みません。）

◇◆◇ 1 行には 1 命令文のみ◇◆◇

一般の BASIC のように、複数の命令文をセミコロンで区切って一行で記述することはできません。

4.4.4 ステップ番号

MELFA-BASIC V では、従来の MELFA-BASIC で使用していた行番号を用いません。代わりに、ステップ番号で扱います。見た目は行番号と同じですが、先頭から 1 つ飛びの番号が自動的に付けられます。ステップ番号は 1 から 32767 までの整数となります。

MELFA-BASIC V では、GoTo, GoSub などの飛び先にステップ番号（行番号）を指定することはできません。指定すると構文エラーとなります。ステップ番号の代わりにラベルを使用してください。

4.4.5 ラベル

ラベルとは分岐先の目印として、独自につけることのできる名前です。
ラベルは、ステップ番号の後に、＊に続けて英大文字、英小文字、数字を使って構成できますが、先頭は必ず英文字とし、16 文字以内で記述します。1 文字のラベルを定義することも可能ですが、予約語となっている文字（F，J，L，P，S，T などは予約語です）がありますので、2 文字以上で判りやすいラベル名をつけることを推奨します。

ラベル名は、入力時の大文字、小文字のまま登録されます。ただし、大文字、小文字の違いは区別せずに扱います。コントローラに登録後、読み出したときは、登録されたラベル名に統一されます。

※ ラベルにできないものを以下に示します。
- 予約語（命令語、関数名等）
- 記号や数字で始まる名前
- 変数名、関数名ですでに使用されている名前
- "_"（アンダースコア）はラベル名の 2 文字目に使用できません。

```
例）1  GoTo  *LBL
          :
       8  *LBL
```

4.4.6 プログラムで使用できる文字の種類

プログラム内で使用できる文字を下表の通りです。ただし、プログラム名、変数名、ラベル名に使用できる文字には制約があります。使用できる文字を○、使用できない文字を ×、制限付きで使用できる文字を△で表してあります。

表 4-5：使用できる文字の一覧

区分	プログラムで使用可能な文字	プログラム名	変数名	ラベル名
英字	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	○	○	○
	a b c d e f g h i j k l m n o p q r s t u v w x y z	×	○	○
数字	0 1 2 3 4 5 6 7 8 9	○	△注 1)	△注 2)
記号	″ ’ & ( ) * + − . , / : ; = < > ? @ ` [¥] ^ { } ~ |	×	×	×
	% & ! # $	×	型指定記号として使用可能	×
	_（アンダースコア）	×	△注 3)	○注 4)
空白	空白文字	×	×	×

注 1）変数名の先頭は英字のみ使用可能です。数字は 2 文字目以降で使用可能です。
注 2）プログラム中のラベル名の先頭は英文字のみ使用可能です。数字は 2 文字目以降で使用可能です。
注 3）2 文字目以降で使用可能です。2 文字目に ″_″（アンダースコア）を使用した変数は外部変数となります。
注 4）ラベル名の 3 文字目以降（先頭の ″*″ は含まず）で使用可能です。

プログラム名についての詳細は、135 ページの「(1) プログラム名」を、変数名についての詳細は、142 ページの「4.4.15 変数」を、ラベル名についての詳細は、136 ページの「4.4.5 ラベル」を参照ください。

4.4.7 特別な意味を持つ文字

(1) 大文字、小文字の区別

MELFA-BASIC V では、英小文字も扱います。

変数名、ラベルは、入力時の大文字、小文字のまま登録されます。ただし、大文字と小文字の区別はされず、先に登録された変数名、ラベル名に統一して扱います。よって、プログラムを読み出したときに、先に登録された名前に変換されます。

また、コメントや文字列データのときは、そのまま登録されます。

命令語やシステム状態変数名についても、大文字と小文字の区別はされず、プログラムを読み出したときに、システム登録された名前に変換されます。

(2) アンダースコア (_)

識別子 (変数の名前) の 2 文字目に _(アンダースコア) をつけることで、その変数をプログラム間にまたがって有効となる外部変数として識別します。詳細は、145 ページの「4.4.22 外部変数」を参照ください。

例) P_Curr、M_01、M_ABC

(3) アポストロフィ (')

アポストロフィ以降の文字を注釈 (コメント) とします。また、文の先頭に付けると Rem 文の代用になります。

```
例 )  100  Mov P1 'TORU              ; TORU が注釈になります。
      150  'BUHIN   TORIDASHI       ; 150 Rem BUHIN TORIDASHI と同じです。
```

(4) アスタリスク (*)

分岐先となるラベル名の先頭に付けます。

```
例 )  200   *KAKUNIN
```

(5) コンマ (,)

命令のパラメータ (引数) や添え字の区切りとして使用します。

```
例 )  P1=(200,  150,  ....)
```

(6) ピリオド (.)

小数点や、位置変数、関節変数などの複数データから、成分データだけを得る場合に使用します。

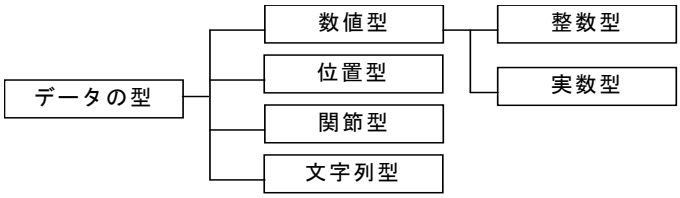
```
例 )  M1=P2.X  ; 位置変数 P2 の X 座標成分だけを数値変数 M1 に代入する。
```

(7) スペース

文字列定数や注釈文に含まれているものは文字としての意味を持ちます。行番号や命令語の直後、データごとの区切りなどに必要です。「4.13 命令の詳細説明」の【書式】では、必要な箇所に〝␣〟でスペースを示しています。

4.4.8 データ型

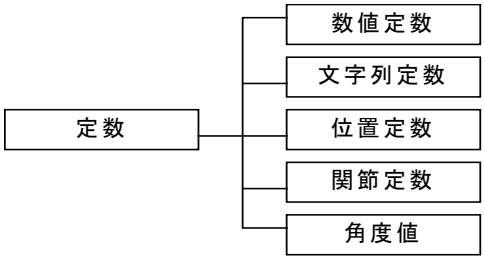
MELFA-BASIC IV では、データの型は数値、位置、関節、文字列の 4 つがあり、それぞれをその「データの型」と呼びます。数値型は実数型と整数型に分類されます。
各データ型に変数、定数があります。



例）数値型　　M1[数値変数]、1[数値定数](整数)、1.5[数値定数](実数)
　　位置型　　P1[位置変数]、(0, 0, 0, 0, 0, 0)(0, 0) [位置定数]
　　関節型　　J1[関節変数]、(0, 0, 0, 0, 0, 0)[関節定数]
　　文字列型　C1$[文字列変数]、"ABC"[文字列定数]

4.4.9 定数

定数には、数値定数、文字列定数、位置定数、関節定数、角度値の 5 種類があります。



4.4.10 数値定数

数値定数の構文を以下に示します。数値定数とは以下のようなものです。

(1) 10 進数

例）1、1.7、-10.5、+1.2E+5(指数表記)
有効範囲　-1.7976931348623157e+308 ～ 1.7976931348623157e+308

(2) 16 進数

例）&H0001、&HFFFF
有効範囲　&H0000 ～ &HFFFF

(3) 2 進数

例）&B0010、&B1111
有効範囲　&B0000000000000000 ～ &B1111111111111111

(4) 定数の型

定数文字に記号をつけることによって定数の型を指定できます。
例）10%(整数)、10000&(長精度整数)、1.0005!(単精度実数)、10.000000003#(倍精度実数)

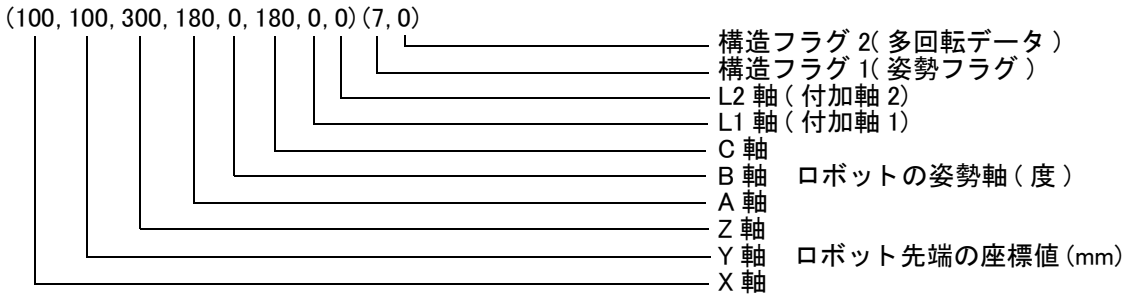
4.4.11 文字列定数

文字列定数は(")ダブルクォーテーション記号で囲んだ文字です。
例）"ABCDEFGHIJKLMN" "123"

◇◆◇文字列は 240 文字まで◇◆◇
入力可能な文字列の長さは行番号、ダブルクォーテーションを含めて 240 文字以内です。
ダブルクォーテーション自体を文字列に含める場合はダブルクォーテーションを 2 つ連続で入力してください。AB"CD という文字列の場合，"AB""CD" と入力します。

4.4.12 位置定数

位置定数の構造を以下に示します。位置定数内には変数は記述できません。



例)

```
P1=(300,  100,  400,  180,  0,  180,  0,  0)(7,  0)
P2=(0,  0,  -5,  0,  0,  0)(0,  0)                    [ 走行軸データなしの場合 ]
P3=(100,  200,  300,  0,  0,  90)(4,  0)              [4 軸の水平多関節ロボの場合 ]
```

(1) 座標・姿勢・付加軸データの形式と意味

【形式】X, Y, Z, A, B, C, L1, L2

【意味】X, Y, Z:　座標データ。直交座標系におけるロボットのハンド先端の位置 (単位は mm)

A, B, C:　姿勢データ。姿勢を示す角度 (単位は deg) 注 1)

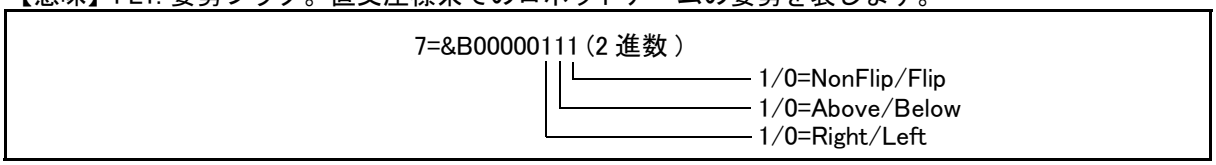
L1, L2 :　付加軸データ。それぞれ付加軸 1、付加軸 2 の座標値 (単位は mm または deg)

注 1)T/B、パソコンサポートソフトウェアは、単位を deg で表示しますが、プログラムでの代入や演算は、単位をラジアンで演算します。

(2) 構造フラグデータの形式と意味

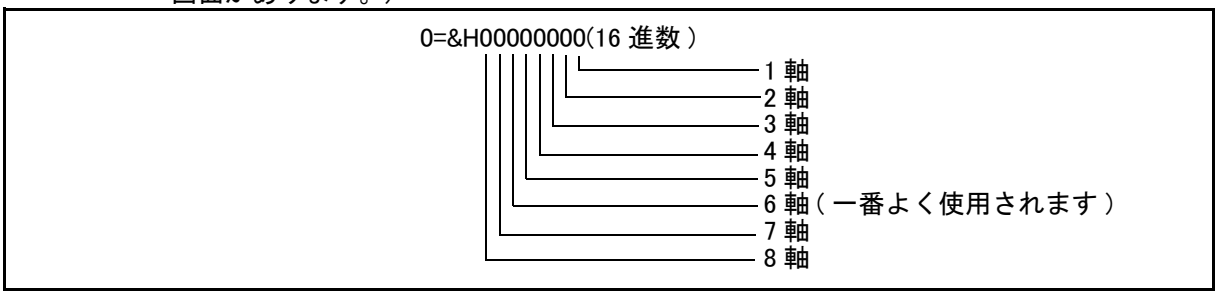
【形式】FL1, FL2

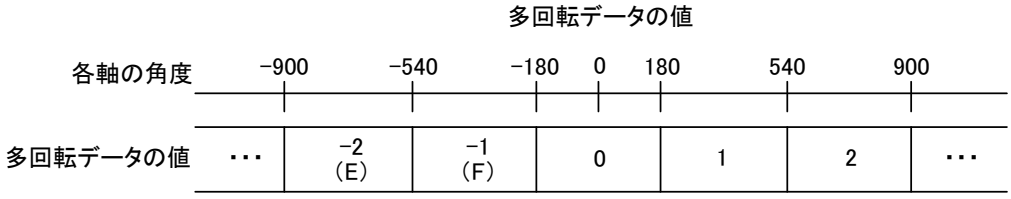
【意味】FL1: 姿勢フラグ。直交座標系でのロボットアームの姿勢を表します。



FL2: 多回転データ。直交座標系で表される位置 (XYZ) と姿勢 (ABC) における各関節軸の回転角度の情報を持ちます。

初期値 =0(範囲は 0 から +4294967295・・1 軸 4 ビット構成で 8 軸分持ちます。1 軸ごとに (-8 ～ 7) パソコンでは 10 進数で表示される画面と軸ごとの回転数を 16 進数で表示される画面があります。)





直交座標系での手首先端軸 (垂直多関節型ロボットでの J6 軸) は、1 回転 (360 度) 回っていても同じ値となります。そのため、この FL2 を用いて区別しています。

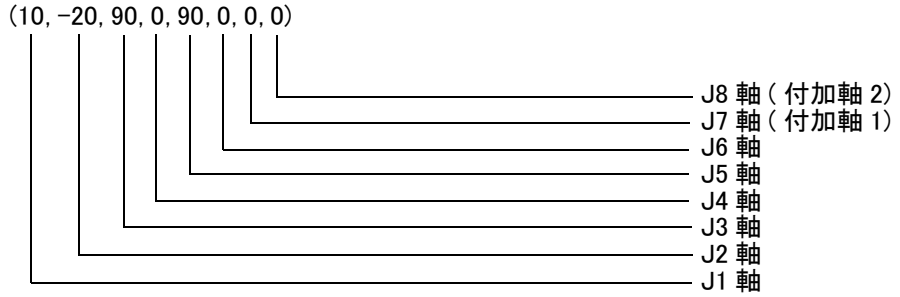
◇◆◇軸数の指定について◇◆◇
①座標・姿勢データは 8 軸分全てを記述する必要はありませんが、省略した場合は以降の軸データは未定義として処理されます。
4 軸ロボット (X, Y, Z, C 軸構成) の場合は、(X,Y,Z,,,C) または (X,Y,Z,0,0,C) と記述します。
②全ての軸を省略する場合は、(,) のように、少なくとも 1 つの ″, ″(コンマ) を入れてください

◇◆◇位置成分データへの変数の使用について◇◆◇
座標・姿勢・付加軸データ、および構造フラグデータのことを位置成分データと呼びます。
位置定数を構成する位置成分データに変数を含むことはできません。

◇◆◇構造フラグデータの省略◇◆◇
構造データを省略したときは初期データ ((7, 0) 機種によって異なります) が適用されます。

4.4.13 関節定数

関節定数の構造を以下に示します。関節定数内には変数は記述できません。



例) 6 軸ロボット　J1=(0, 10, 80, 10, 90, 0)
6 軸 + 付加軸　J1=(0, 10, 80, 10, 90, 0, 10, 10)
5 軸ロボット　J1=(0, 10, 80, 0, 90, 0)
5 軸 + 付加軸　J1=(0, 10, 80, 0, 90, 0, 10, 10)
4 軸ロボット　J1=(10, 20, 90, 0)
4 軸 + 付加軸　J1=(10, 20, 90, 0, , , 10, 10)

(1) 各軸データの形式と意味

【形式】J1,J2,J3,J4,J5,J6,J7,J8
【意味】J1 から J6: ロボット各軸データ (単位は mm または deg)
J7,J8: 付加軸データ、省略可能です (単位は mm または deg　パラメータの設定によります。)
水平多関節型ロボットの J3 軸が直動軸の場合の単位は角度ではなく mm になります。

◇◆◇関節成分データへの変数の使用について◇◆◇
各軸データのことを関節成分データと呼びます。
関節定数を構成する関節成分データに変数を含むことはできません。

4.4.14 角度値

角度値は〝ラジアン〟ではなく、〝度〟によって表現する時に使用します。

100Deg　と記述すると角度値になり、三角関数の引数などに使用できます。

例)Sin(90Deg).....90 度の正弦を示します。

4.4.15 変数

変数名は最大 16 文字で作成します。英字 1 文字の変数名を使用することは可能ですが、予約語となっている文字 (F，J，L，P，S，T など) がありますので、2 文字以上の判り易い名前をつけることを推奨します。

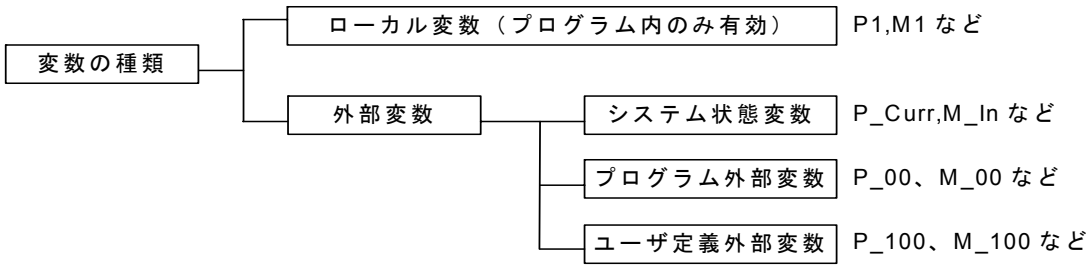
変数の型は、数値型、文字列型、位置型、関節型、入出力型があります。それぞれをその「変数の型] と呼びます。変数の型は識別子 (変数の名前) の先頭文字で決まります。

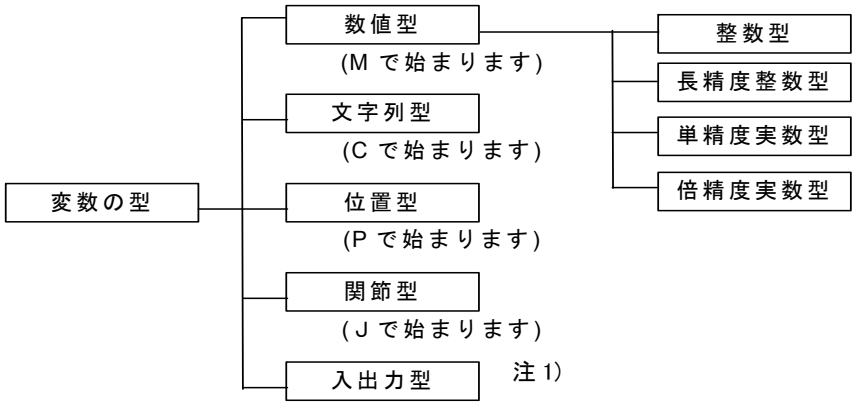
なお、数値型は、整数型、長精度整数型、単精度実型、倍精度実数型に分かれます。

また、データの有効範囲としては、以下の 2 種類があります。

① 1 つのプログラム内でのみ有効なローカル変数

②プログラム間にまたがって有効なロボット (システム) 状態変数、プログラム外部変数、ユーザ定義外部変数 (変数名の第 2 文字目に _ の付いた外部変数。145 ページの「4.4.22 外部変数」参照。)





注 1) 識別子はロボット (システム) 状態変数で決まっているもの (M_In,M_Out など) と、Def IO 命令でプログラムで宣言するものとがあります。

◇◆◇変数は初期化されません◇◆◇

変数は、生成された時やプログラムロードまたはリセットにおいて 0 クリアされることはありません。

4.4.16 数値変数

数値変数は P，J，C の文字以外の文字で始まる変数です。MELFA-BASIC IV では M を先頭につけて説明している場合が多いです。M は mathematics(数学) の頭文字です。

例) M1=100
M2!=-1.73E+10
M3#=0.123
ABC=1

1) 数値型後置子にて変数の型を定義することが可能です。省略した場合は単精度実数型となります。
表 4-6 : 数値変数の型とデータ範囲

数値型後置子	型	範囲	備考
%	整数型	-32768 ～ 32767	
&	長精度整数型	-2147483648 ～ 2147483647	
!	単精度実数型	-3.40282347e+38 ～ 3.40282347e+38	注) e は 10 のべき乗を表します。
#	倍精度実数型	-1.7976931348623157e+308 ～ 1.7976931348623157e+308	

2) 登録済みの変数にあとから数値型後置子をつけることはできません。新規プログラムの作成時に最初に記述する時点で記号を書いてください。
3) 単精度 = 倍精度の実行において、値がオーバーした場合はエラーとなります。

4.4.17 文字列変数

文字列変数は "C"(または "c") で始まり最後に "$" をつけます。Def Char 命令で定義した場合は "C"(または "c") 以外の文字から始まる変数名をつけることも可能です。

例) C1$="ABC"
CS$=C1$
Def Char MOJI
MOJI="MOJIMOJI"

4.4.18 位置変数

位置変数は "P"(または "p") で始まる変数です。Def Pos 命令で定義した場合は "P"(または "p") 以外の文字から始まる変数名をつけることも可能です。
位置変数の成分データを参照できます。変数名のあとに "." と成分名 "X" などをつけます。

P1.X、 P1.Y、 P1.Z、 P1.A、 P1.B、 P1.C、 P1.L1、 P1.L2

角度成分の ABC の単位は (rad) です。度に変換するには Deg 関数を使用してください。

例) P1=PORG
Dim P3(10)
M1=P1.X (単位 mm)
M2=Deg(P1.A) (単位 DEG)
Def Pos L10
Mov L10

4.4.19 関節変数

関節変数は "J"(または "j") で始まる変数です。Def Jnt 命令で定義した場合は "J"(または "j") 以外の文字から始まる変数名をつけることも可能です。

関節変数の成分データを参照できます。変数名のあとに "." と成分名 "J1" などをつけます。

JDATA.J1、 JDATA.J2、 JDATA.J3、 JDATA.J4、
JDATA.J5、 JDATA.J6、 JDATA.J7、 JDATA.J8

成分データの単位は (rad) です。度に変換するには Deg 関数を使用してください。

例) JSTART=(0, 0, 90, 0, 90, 0, 0, 0)

```
JDATA=JSTART
Dim J3(10)
M1=JDATA.J1        ( 単位   RAD)
M2=Deg(JDATA.J2) ( 単位   DEG)
Def Jnt K10
Mov K10
```

4.4.20 入出力変数

入出力変数は以下の種類があります。これはロボット (システム) 状態変数であらかじめ用意されています。

入出力変数名	説明
M_In	入力信号のビット参照用
M_Inb	入力信号のバイト (8 ビット) 参照用
M_Inw	入力信号のワード (16 ビット) 参照用
M_Out	出力信号のビット参照 / 代入用
M_Outb	出力信号のバイト (8 ビット) 参照 / 代入用
M_Outw	出力信号のワード (16 ビット) 参照 / 代入用

ロボット (システム) 状態変数の 315 ページの「M_In/M_Inb/M_In8/M_Inw/M_In16」、325 ページの「M_Out/M_Outb/M_Out8/M_Outw/M_Out16」および 307 ページの「M_DIn/M_DOut」を参照ください。

4.4.21 配列変数

数値変数、文字列変数、位置変数、関節変数は配列が使用できます。変数の添字部分で配列要素を指定します。配列変数のは Dim 命令で記述します。最大 3 次元配列まで可能です。

例) 配列変数の定義例

```
Dim M1(10)        単精度実数型
Dim M2%(10)       整数型
Dim M3&(10)       長精度整数型
Dim M4!(10)       単精度実数型
Dim M5#(10)       倍精度実数型
Dim P1(20)
Dim J1(5)
Dim ABC(10, 10, 10)
```

配列の添字は 1 からです。

入出力信号のロボット (システム) 状態変数の中で入出力信号用の変数 (M_In、M_Out など) だけは添え字は 0 からになります。

空きメモリのサイズにより領域が確保可能かが決まります。

4.4.22 外部変数

外部変数とは、その識別子（変数の名前）の 2 文字目に″_″（アンダースコア）をつけた変数で、その値は複数のプログラム間にまたがって有効となります（ユーザ定義外部変数はベースプログラムに登録する必要があります）。このため、マルチタスクのプログラム間でデータの受け渡しをする場合に有効です。外部変数も、139 ページの「4.4.8 データ型」に示したものと同様に数値、位置、関節、文字の 4 種類の型があります。外部変数には以下の 3 種類があります。

表 4-7：外部変数の種類

外部変数	説明	例
プログラム外部変数	その名前があらかじめ決まっています。	P_01, M_01, P_100(1) など
ユーザ定義外部変数	ユーザにて任意に名前を決められます。 ユーザベースプログラムにて Def Pos, Def Jnt, Def Char, Def Inte/Long/Float/Double 命令にて宣言します。	P_GENTEN, M_MACHI
ロボット（システム）状態変数	ロボット（システム）状態変数はシステムで管理され、その名前に対する内容（用途）があらかじめ決まっています。	M_In, M_Out, P_Curr, M_PI など

4.4.23 プログラム外部変数

コントローラに最初から用意されているプログラム外部変数を表 4-8 に示します。表に示すように、変数の名前は決まっていますが、その用途はユーザで決められます。

表 4-8：プログラム外部変数

データ型	変数名	個数	備考
位置	P_00 ～ P_39	40	
位置配列（要素数 10）	P_100( ) ～ P_109( )	10	配列の要素は 1 次元でご使用ください。
関節	J_00 ～ J_39	40	
関節配列（要素数 10）	J_100( ) ～ J_109( )	10	配列の要素は 1 次元でご使用ください。
数値	M_00 ～ M_39	40	変数の型は倍精度実数です。
数値配列（要素数 10）	M_100( ) ～ M_109( )	10	配列の要素は 1 次元でご使用ください。変数の型は倍精度実数です。
文字列	C_00 ～ C_39	40	
文字列配列（要素数 10）	C_100( ) ～ C_109( )	10	配列の要素は 1 次元でご使用ください。

4.4.24 ユーザ定義外部変数

前記のプログラム外部変数のみでは足りない場合、独自の名前の付いた変数を定義したい場合はユーザベースプログラムを利用することによってプログラム外部変数をユーザにて定義することが可能です。

ユーザ定義外部変数を使うまでの手順
1) 最初にユーザベースプログラムを作成します。変数の 2 文字目には "_" をつけます。 2) パラメータ「PRGUSR」にプログラム名を設定し電源を再投入します。 3) ユーザ定義外部変数を使用した通常のプログラムを作成します。

(1) ユーザベースプログラム[注)] 中の Def 文で識別子の 2 文字目に "_"(アンダースコア) を付けた変数を定義することによってその変数はユーザ定義外部変数となります。
(2) ユーザベースプログラムは実行する必要はありません。
(3) ユーザベースプログラムの中には変数の宣言の行のみ記述してください。
(4) ユーザベースプログラムにて配列変数を定義し、外部変数として使用する場合は、使用するプログラム側でも再度 Dim 命令にて配列宣言する必要があります。ローカル変数 (プログラム内でのみ有効な変数) は再宣言の必要はありません。

例) ユーザ定義外部変数の使用例

メインプログラム (プログラム名　1) 側

```
1  Dim P_200(10)                    ' 外部変数の再宣言
2  Dim M_200(10)                    ' 外部変数の再宣言
3  Mov P_100(1)
4  If M_200(1)=1 Then Hlt
5  M1=1                             ' ローカル変数
```

ユーザベースプログラム (プログラム名　100) 側

```
1  Def Pos P_900, P_901, P_902, P_903
2  Dim P_200(10)                    ' 使用するプログラム側でも再度宣言が必要。
3  Def Inte M_100
4  Dim M_200(10)                    ' 使用するプログラム側でも再度宣言が必要。
```

パラメータ名	値
PRGUSR	100

4.4.25 ユーザベースプログラムの作成

注)

◇◆◇ユーザベースプログラムとは？◇◆◇

ユーザベースプログラムは、ユーザ定義外部変数を使う場合、その変数を定義するために使用しますが、本プログラムは実行する必要はありません。宣言文を記述したプログラムを作成しパラメータ「PRGUSR」に登録するのみです。パラメータを変更したあと電源を一度入れ直してください。

◇◆◇パソコンサポートソフトウェアでユーザーベースプログラムを新規に登録する方法◇◆◇

最初にパソコンサポートソフトウェアからコントローラへの命令のみを書込み、次に位置データのみを書き込みます。

ユーザベースプログラムは、通常のプログラム同様にティーチングボックス、パソコンサポートソフトウェアのいずれを用いても作成できますが、パソコンサポートソフトウェアで作成の場合は以下の手順でおこなってください。

1) ユーザベース用に作成したプログラムをパソコンに保存します。
2) パソコンサポートソフトウェアのプログラム編集からプログラム管理を起動します。
3) プログラム管理の転送元に 1) で作成したプログラムを指定し、転送先をロボットにし、「コピー」操作をおこないます。このとき、「位置変数」のチェックを外し、「命令文」だけチェックされている状態にし実行します。
4) コピーが完了したら、再び 3) の操作をおこない、今度は、「命令文」のチェックを外し、「位置変数」だけチェックされている状態にして実行します。
5) プログラム管理でプログラムを削除し再登録する場合も、先にユーザーベースプログラムをコントローラへ書き込んでください。

4.5 ロボットの座標系解説

4.5.1 ロボットの座標系について

ロボットの座標系には、以下の 4 つがあります。

①ワールド座標系：原点は ※1

ロボットの現在位置[注 1)] を表す基準となる座標系。

②ベース座標系：原点は ※2(ロボット底面の J1 軸回転中心軸)

ロボットのベースの取付面を基準として定められる座標系。ワールド座標系からみたロボットの据付中心位置 (ベース変換データ) をパラメータ：MEXBS への設定、または Base 命令の実行で設定します。

初期設定では、ベース変換データは 0 のためワールド座標系はベース座標系に一致しています。

③メカニカルインタフェース座標系：原点は ※3(ツール取付面上の J6 軸回転中心軸)

ロボットのメカニカルインタフェース面を基準として定められる座標系。

④ツール座標系：原点は ※4

ロボットのメカニカルインタフェース面に取り付けられたツール (ハンド) に定められる座標系。メカニカルインタフェース座標系との関係はツールデータ (パラメータ：MEXTL の設定、または Tool 命令の実行) により決まります。

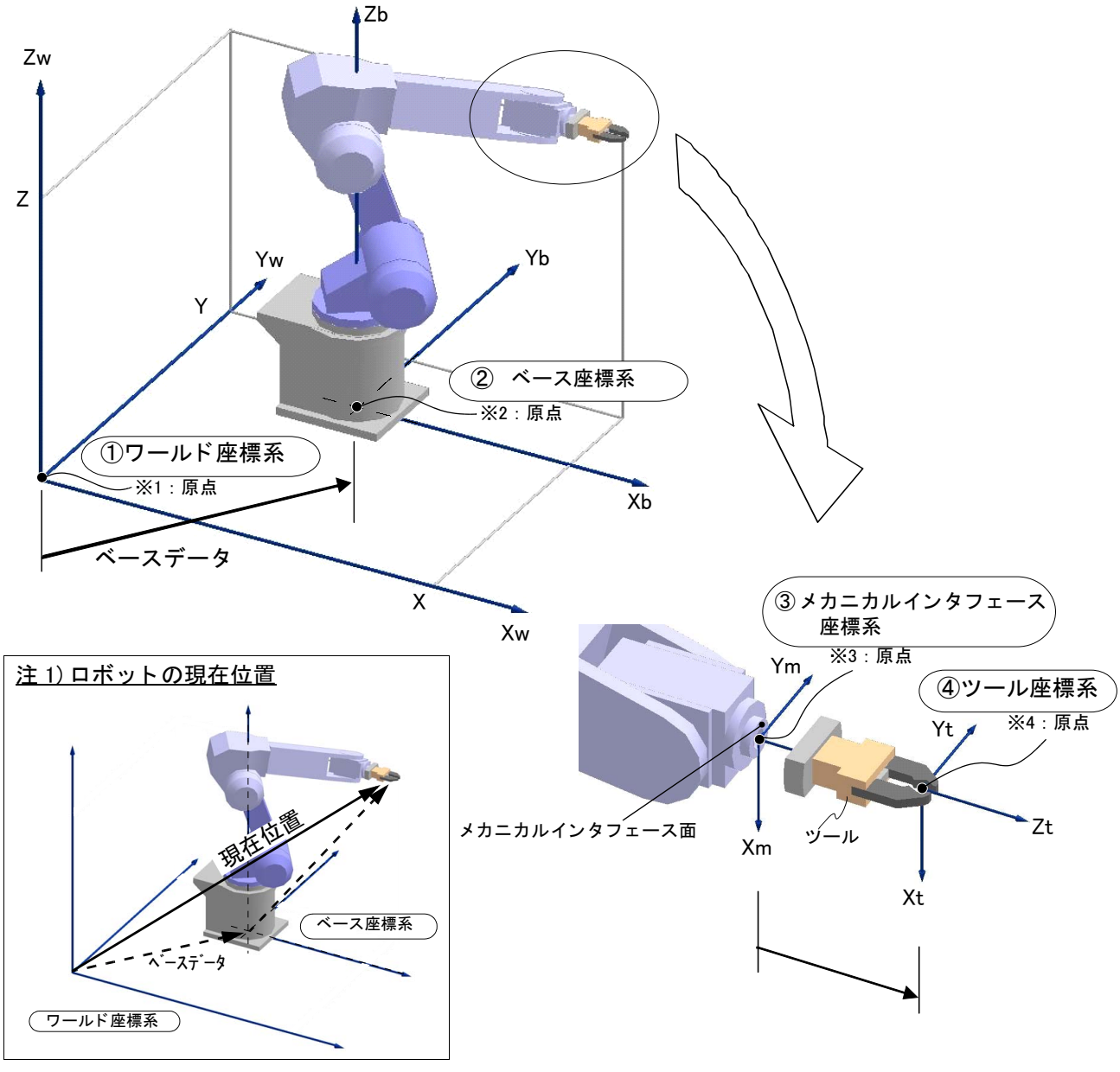


図 4-3：ロボットの座標系

4.5.2 ベース変換について

必要に応じてワールド座標系を作業台の基準位置や、ワークの基準位置に移動させることができます。この機能を活用すると、ロボットの現在位置が作業台やワーク基準とした位置で扱われるため、その基準に対するロボットの動作位置関係が同一となる作業グループが複数個存在する場合に、作業グループのすべての動作を教示する必要はなく、ワールド座標系の変更のみで同一作業(動作)が可能となります。このワールド座標系の変更をベース変換といい、パラメータ:MEXBS へのベース変換データ(座標値)の設定、または Base 命令の実行でおこないます。

設定するベース変換データは、新しく設定するワールド座標系からみたベース座標系の原点位置データとします。このため、ロボットの現在位置を使用して設定する場合(Fram 関数など)は、位置データを逆変換して設定します。(例:Base Inv(P1))
ワーク座標系パラメータ(WK1CORD ～ WK8CORD)を Base 命令で指定する場合は、内部の処理で逆変換をおこないますので、お客様での逆変換処理は不要です。(例:Base1 ～ Base8)

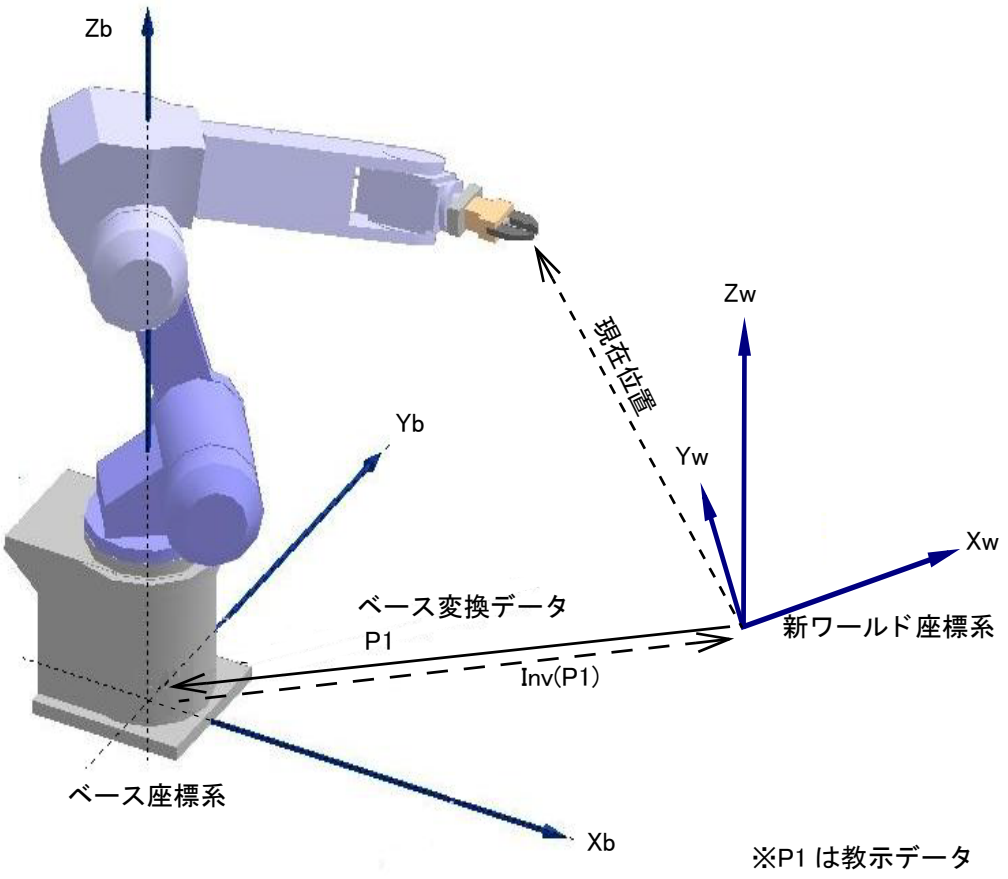


図 4-4:ベース変換

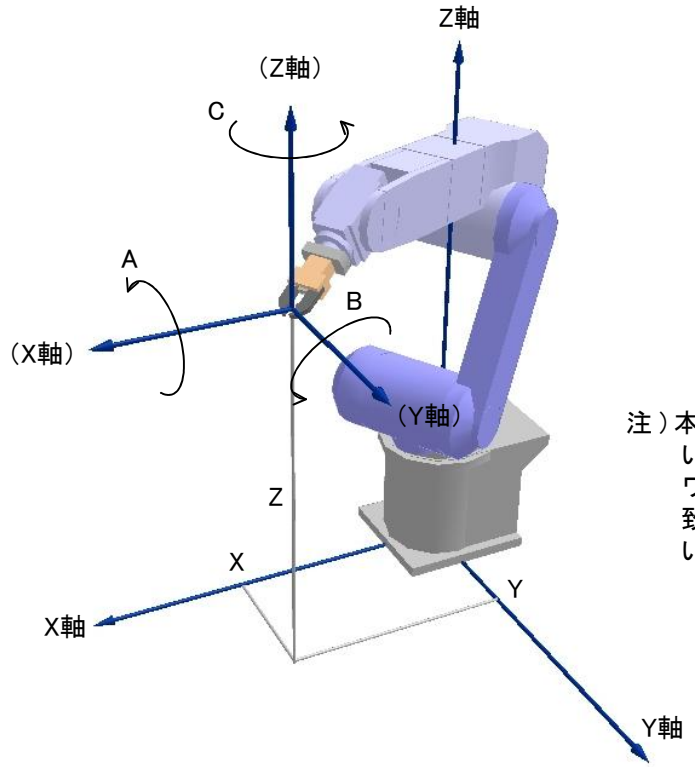
⚠注意

ベース変換をおこなうとロボットの現在位置は新しく設定されたワールド座標系を基準とした値に変わります。移動命令で指定された目的位置も新しく設定されたワールド座標系内の位置として扱われます。
したがって、それまでの教示データはそのままでは使えなくなる場合があります。教示時の座標系と新しく設定された座標系が同一である必要があります。
座標系が一致していないと、ロボットは思わぬ位置へ移動し、物損や人身事故の原因となります。
ベース変換機能をご使用になる場合は、変換するベース座標系と教示位置の関係をしっかりと管理して、ロボットを正しく動作させ、ベース変換機能を効果的にご活用いただきますようお願いします。

4.5.3 位置データについて

ロボットの位置データとは、ハンド先端(ツール設定されていないときは、メカニカルインタフェース中心)の位置(X, Y, Z)と姿勢(A, B, C)の6個のデータ、および構造フラグを合わせたものをいいます。各データの値は、ワールド座標系基準となります。

【意味】 X, Y, Z：座標データ。ワールド座標系におけるロボットのハンド先端の位置(単位はmm)
A, B, C：姿勢データ。姿勢を示す角度(単位はdeg)
A → X軸回りの回転角
B → Y軸回りの回転角
C → Z軸回りの回転角



注)本図は、ベースデータが設定されていない状態、つまり、ロボットのワールド座標系とベース座標系が一致している場合を想定して描かれています。

姿勢角の定義：
A, B, Cは、ハンド先端(ツール設定されていないときは、フランジ中心)の座標系の姿勢を表したもので、それぞれワールド座標系のX軸、Y軸、Z軸回りの回転角を示します。各座標軸の+方向に対し右ねじの回転方向が+となります。また、回転する順番は決められており、Z軸回り→Y軸回り→X軸回りの順に回転させて、計算(制御)しています。

4.5.4 ツール座標系(メカニカルインタフェース座標系)について

ロボットの制御点をロボットに取り付けられたハンド先端にする場合にツールデータを設定する必要があります。ツールデータはフランジに設定されているメカニカルインタフェース座標系を基準に、ツール先端位置を定義したものです。したがって、まずメカニカルインタフェース座標系について説明します。

ツール座標系を理解していただくために、ここでは垂直 6 軸ロボットを例に説明しますが、その他の機種(水平関節型ロボットなど)については、432 ページの「5.6 標準ツール座標について」を参照ください。

(1) メカニカルインタフェース座標系

図 4-5 に示すように、フランジ中心を原点とする座標系をメカニカルインタフェース座標系と呼びます。メカニカルインタフェース座標系の X 軸を Xm、Y 軸を Ym、Z 軸を Zm と表記します。

Zm はフランジ中心を通り、フランジ面に垂直な軸です。フランジ面から外に向かう方向が + となります。Xm、Ym はフランジ面内に設定されます。フランジ中心と、位置決め用ピン穴を結ぶ線が Xm 軸になります。Xm 軸の + 方向は、中心からピン穴と反対方向です。

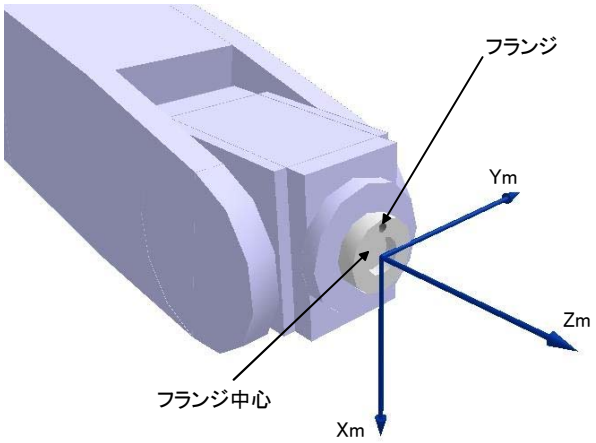


図 4-5 : メカニカルインタフェース座標系

フランジが回転すると、メカニカルインタフェース座標系も回転します。(図 4-6)

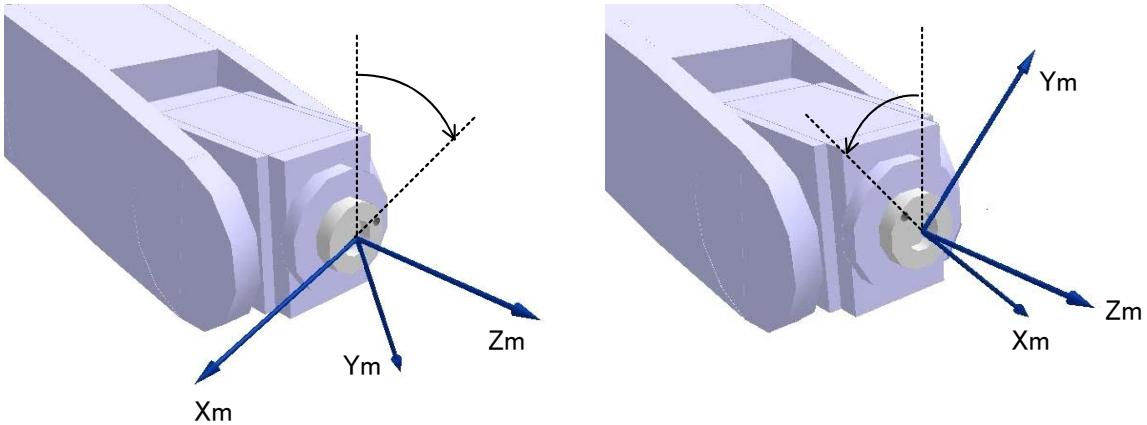


図 4-6 : フランジ回転とメカニカルインタフェース座標系

(2) ツール座標系

ツール座標系は、ハンド先端(ハンド制御点)に定義された座標系で、メカニカルインタフェース座標系の原点をハンド先端(制御点)にシフトさせ、任意の回転成分を加えたものです。
ツール座標系の X 軸を Xt、Y 軸を Yt、Z 軸を Zt と表記します。

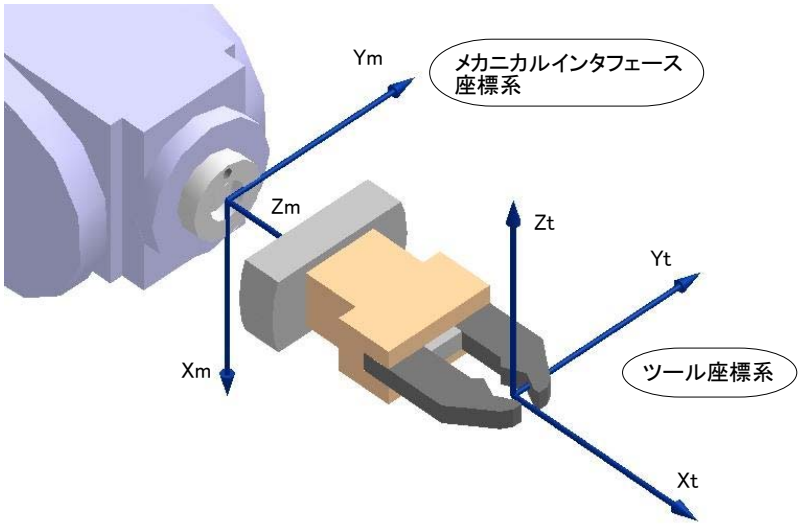


図 4-7 : メカニカルインタフェース座標系とツール座標系

ツールデータは、位置データと同じ成分を持っています。

X, Y, Z : シフト量。メカニカルインタフェース座標系原点からツール座標系原点までの移動量。
(単位は mm)

A, B, C : 座標軸の回転角度。(単位は deg)

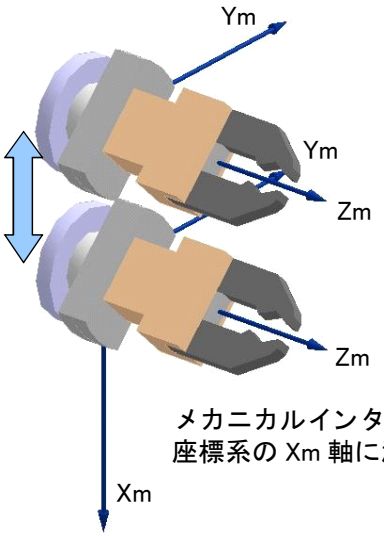
A → X 軸回りの回転角
B → Y 軸回りの回転角
C → Z 軸回りの回転角

(3) ツール座標系使用の効果

1) ジョグ動作、教示操作時
ツールジョグモードでロボットを動作させると、ハンド面の方向にしたがって動作させることができます。対象ワークに対するハンド姿勢の調整や、把持しているワークの姿勢を変更しやすくなります。

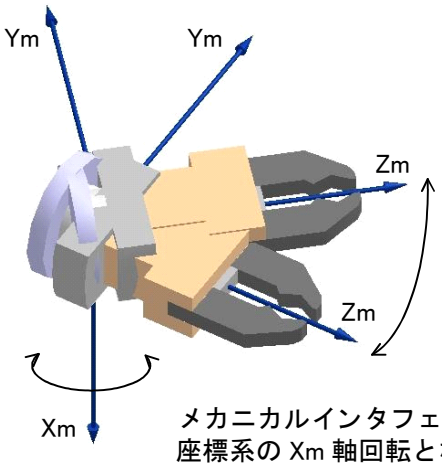
ツールデータが設定されていない場合

X 方向移動



メカニカルインタフェース座標系の Xm 軸に沿った動作

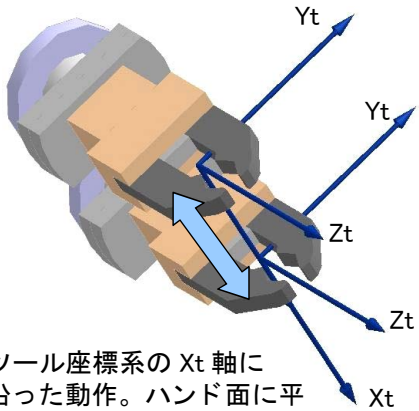
A 方向移動



メカニカルインタフェース座標系の Xm 軸回転となり、ハンド先端は大きく移動する

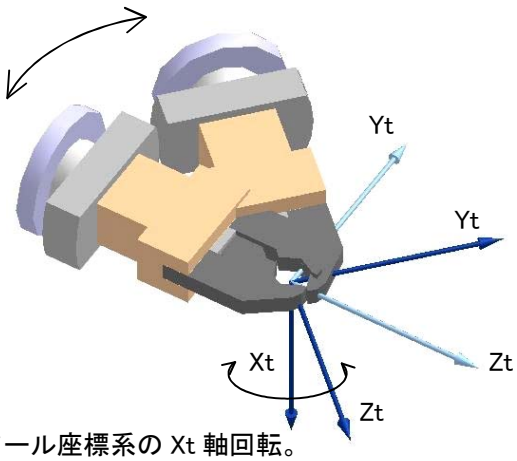
ツールデータが設定されている場合

X 方向移動



ツール座標系の Xt 軸に沿った動作。ハンド面に平行 / 垂直に動作し、ワークの向きと合わせることができる。

A 方向移動



ツール座標系の Xt 軸回転。ハンド先端を中心とする回転動作となり、ワークの位置をずらさず姿勢変更が可能となる。

図 4-8：ツールデータ有無によるツールジョグ動作

2) 自動運転時

移動命令では、近接距離 / 離脱距離を指定し、ワーク取り出し、搬送時の動作を簡単に設定することができます。近接 / 離脱方向はツール座標系の Z 方向です。

図 4-9 のように、ワーク搬送位置上空 50mm の位置に移動する場合は、

Mov P1,50

のように記述します。P1 の Z 軸方向 (ツール座標系) で +50mm の位置へ移動することを意味します。

ツール座標系の Z 方向をワークの向きや動作状態に合わせて設定しておくと作業性が向上します。

図 4-9 の例では、ハンドを横向きにしてワークの挿入 / 引き抜きをおこなうため、ツール座標系の Z 軸方向をワークの向きに合わせています。

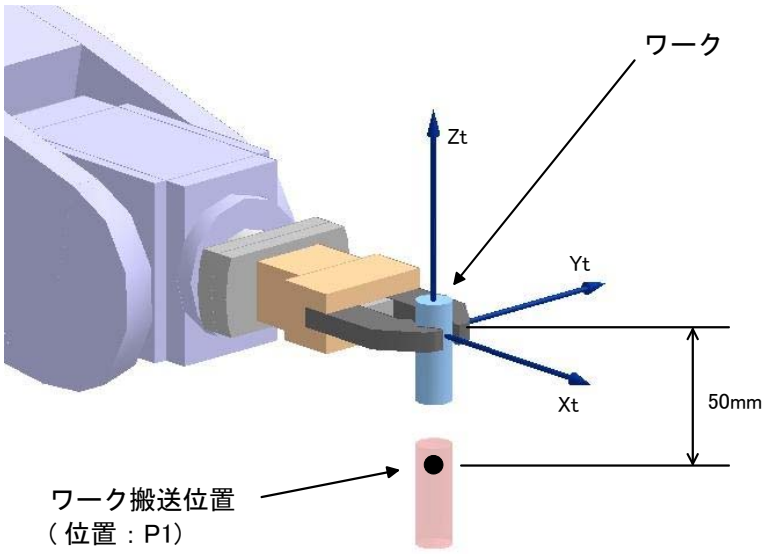


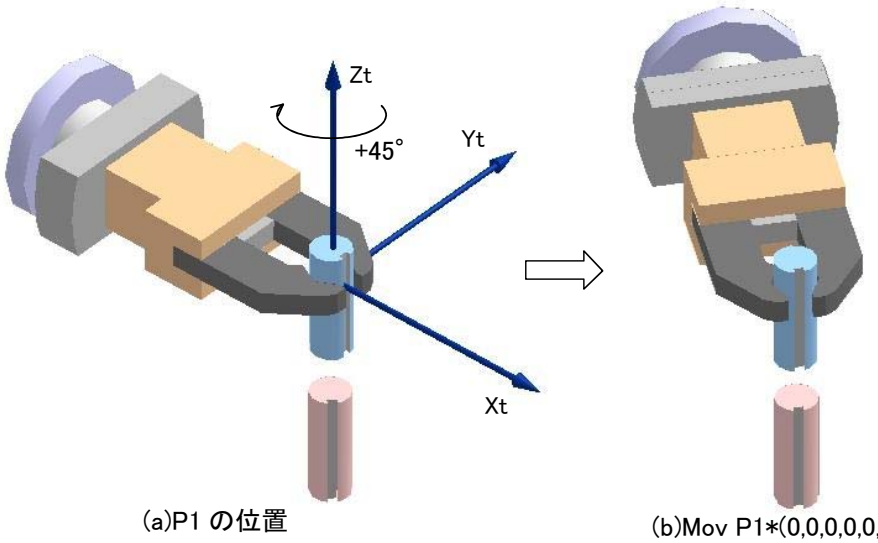
図 4-9 : 近接 / 離脱動作

ワークの位相合わせなどのように、ワークの姿勢変更を行う場合も、ツールデータが設定されていると便利です。

図 4-10 のように、ワークの中心軸まわりに回転させて位相を合わせる場合、

Mov P1*(0,0,0,0,0,45)

のように記述します。「*(0,0,0,0,0,45)」は、「*」でツール座標系上での位置演算を意味し、(X,Y,Z,A,B,C) の C を 45° 回転させることを意味します。C は Z 軸回りの回転ですから、ロボットの動作としては、P1 の Z 軸回り (ツール座標系の Zt 軸まわり) に 45° 回転することになります。



(a)P1 の位置　　(b)Mov P1*(0,0,0,0,0,45) の位置)

図 4-10 : ツール座標系での回転動作

4.6 ロボット(システム)状態変数

あらかじめ用意されているロボットの状態変数を表 4-9 に示します。表に示すように、変数の名前とその用途があらかじめ決まっています。
これらの変数を使用することでロボットの状態を調べたり、状態を変化させることができます。

表 4-9 : ロボット (システム) 状態変数

No	変数名称	配列要素数注 1)	内容 .	属性注 2)	データの型、単位	ページ
1	P_Curr	メカ番号 (1 ～ 3)	現在位置(直交)	R	位置型	358
2	J_Curr	メカ番号 (1 ～ 3)	現在位置(関節)	R	関節型	293
3	J_ECurr	メカ番号 (1 ～ 3)	現在エンコーダパルス位置	R	関節型	297
4	J_Fbc	メカ番号 (1 ～ 3)	サーボからのフィードバック値を元に生成した関節位置	R	関節型	298
5	J_AmpFbc	メカ番号 (1 ～ 3)	電流のフィードバック値	R	関節型	298
6	P_Fbc	メカ番号 (1 ～ 3)	サーボからのフィードバック値を元に生成した直交位置	R	位置型	361
7	M_Fbd	メカ番号 (1 ～ 3)	指令位置とフィードバック位置の距離	R	整数型, mm	313
8	M_CmpDst	メカ番号 (1 ～ 3)	コンプライアンス機能実行時の、指令値と実際の位置のずれ量	R	単精度実数型, mm	303
9	M_CmpLmt	メカ番号 (1 ～ 3)	コンプライアンスモードの指令値がリミットを越えようとしてエラーとなったとき、割り込み処理を使ってエラー状態から復旧させる場合に使用。	R	整数型	304
10	P_Tool	メカ番号 (1 ～ 3)	現在指定されているツール変換データ	R	位置型	362
11	P_Base	メカ番号 (1 ～ 3)	現在指定されているベース変換データ	R	位置型	354
12	P_NTool	メカ番号 (1 ～ 3)	システム初期値(ツール変換データ)	R	位置型	362
13	P_NBase	メカ番号 (1 ～ 3)	システム初期値(ベース変換データ)	R	位置型	354
14	M_Tool	メカ番号 (1 ～ 3)	ツール番号(1～16)	RW	整数型	339
15	J_ColMxl	メカ番号 (1 ～ 3)	推定トルクと実トルクの差	R	関節型、%	294
16	M_ColSts	メカ番号 (1 ～ 3)	衝突検知状態(1:衝突中、0:それ以外)	R	整数型	305
17	P_ColDir	メカ番号 (1 ～ 3)	衝突時の動作方向	R	位置型	356
18	P_CordR	メカ番号 (1 ～ 3)	干渉回避機能で共通座標から自ロボットベース座標までの位置を返します。	R	位置型	357
19	P_CurrR	メカ番号 (1 ～ 3)	干渉回避機能で共通座標から見た自ロボットの現在位置を返します。	R	位置型	359
20	M_Cavsts	メカ番号 (1 ～ 3)	干渉回避機能で干渉を検出したロボットCPU番号を返します。	RW	整数型	302
21	P_CavDir	メカ番号 (1 ～ 3)	干渉回避機能で干渉を検出した時のロボットの動作方向を返します。	R	位置型	355
22	M_OPOvrd	なし	操作パネルの速度オーバーライド(0～100%)	R	整数型、%	319
23	M_Ovrd	スロット番号 (1 ～ 32)	現在指定されているﾌﾟﾛｸﾞﾗﾑ内ｵｰﾊﾞﾗｲﾄﾞ(0～100%)	R	整数型、%	319
24	M_JOvrd	スロット番号 (1 ～ 32)	現在指定されている関節ｵｰﾊﾞﾗｲﾄﾞ(0～100%)	R	整数型、%	319
25	M_NOvrd	スロット番号 (1 ～ 32)	システム初期値(M_OVRDの初期値)(%)	R	単精度実数型、%	319
26	M_NJovrd	スロット番号 (1 ～ 32)	システム初期値(M_JOVRDの初期値)(%)	R	単精度実数型、%	319
27	M_Wupov	メカ番号 (1 ～ 3)	暖機運転オーバーライド(50～100%)	R	単精度実数型、%	346
28	M_Wuprt	メカ番号 (1 ～ 3)	暖機運転状態を解除するまでの時間(秒)		単精度実数型、秒	347
29	M_Wupst	メカ番号 (1 ～ 3)	再び暖機運転状態となるまでの時間(秒)		単精度実数型、秒	348
30	M_Ratio	スロット番号 (1 ～ 32)	現在動作中の目的位置までの到達率(%)	R	整数型、%	330
31	M_RDst	スロット番号 (1 ～ 32)	現在動作中の残距離(X,Y,Zの3次元のみを考慮 mm)	R	単精度実数型, mm	331
32	M_Spd	スロット番号 (1 ～ 32)	現在指定されている速度(直線/円弧補間にのみ有効)	R	単精度実数型、mm/s	334
33	M_NSpd	スロット番号 (1 ～ 32)	システム初期値(M_SPDの初期値)(mm/s)	R	単精度実数型、mm/s	334
34	M_RSpd	スロット番号 (1 ～ 32)	現在の指令速度(mm/s)	R	単精度実数型、mm/s	334
35	M_Acl	スロット番号 (1 ～ 32)	現在指定されている加速率(%)	R	単精度実数型、%	299
36	M_DAcl	スロット番号 (1 ～ 32)	現在指定されている減速率(%)	R	単精度実数型、%	299
37	M_NAcl	スロット番号 (1 ～ 32)	システム初期値(M_ACLの初期値)(%)	R	単精度実数型、%	299
38	M_NDacl	スロット番号 (1 ～ 32)	システム初期値(M_DACLの初期値)(%)	R	単精度実数型、%	299
39	M_AclSts	スロット番号 (1 ～ 32)	現在の加減速状態0=停止中 1=加速中 2=定速中 3=減速中	R	整数型	299
40	M_SetAdl	軸数 (1 ～ 8)	各軸の加減速時間比率を指定します。(%)	RW	単精度実数型、%	332
41	M_LdFact	軸数 (1 ～ 8)	各軸のサーボモータの負荷率(%)	R	単精度実数型、%	321
42	M_Run	スロット番号 (1 ～ 32)	運転中状態(1:運転中 0:運転中でない)	R	整数型	331
43	M_Wai	スロット番号 (1 ～ 32)	中断中状態(1:中断中 0:中断中でない)	R	整数型	345

No	変数名称	配列要素数注1)	内容 .	属性注2)	データの型、単位	ページ
44	M_Psa	スロット番号(1 ～ 32)	指定したタスクスロットにてプログラム選択が可能かを返します。(1:選択可、0:選択不可　中断中時)	R	整数型	329
45	M_Cys	スロット番号(1 ～ 32)	サイクル運転状態(1:サイクル運転中、0:サイクル運転中でない)	R	整数型	306
46	M_Cstp	なし	サイクル停止動作状態(1:サイクル停止中 0:サイクル停止中でない)	R	整数型	306
47	C_Prg	スロット番号(1 ～ 32)	実行プログラム名	R	文字列型	291
48	M_Line	スロット番号(1 ～ 32)	現在実行している行番号	R	整数型	320
49	M_SkipCq	スロット番号(1 ～ 32)	最後に実行したSKIPコマンドを含んだ行の実行を行った結果、SKIPした場合は1、SKIPしなかった場合0が入ります	R	整数型	333
50	M_BrkCq	なし	BREAK命令結果(1:BREAK、0:なし)	R	整数型	301
51	M_Err	なし	エラー中(1:エラー発生中　0:エラー発生中でない)	R	整数型	311
52	M_ErrLvl	なし	エラーレベルを読み出す ・S/WﾊﾞｰｼﾞｮﾝR1d以前(SQｼﾘｰｽﾞ)/S1d以前(SDｼﾘｰｽﾞ) ｴﾗｰなし/警告/ﾛｰﾚﾍﾞﾙ/ﾊｲﾚﾍﾞﾙ=0/1/2/3 ・S/WﾊﾞｰｼﾞｮﾝR1d以降(SQｼﾘｰｽﾞ)/S1d以降(SDｼﾘｰｽﾞ) ｴﾗｰなし/警告/ﾛｰﾚﾍﾞﾙ/ﾊｲﾚﾍﾞﾙ/警告1/ﾛｰﾚﾍﾞﾙ1/ﾊｲﾚﾍﾞﾙ1=0/1/2/3/4/5/6	R	整数型	311
53	M_Errno	なし	エラー番号を読み出す。	R	整数型	311
54	M_Svo	メカ番号(1 ～ 3)	サーボモータ電源ON中(1:サーボ電源ON中　0:サーボON中でない)	R	整数型	337
55	M_Uar	メカ番号(1 ～ 3)	ビットデータ(1:ユーザ定義領域内、0:ユーザ定義領域外) (ビット0:エリア1 ～ビット15:エリア16)	R	整数型	340
56	M_Uar32	メカ番号(1 ～ 3)	ビットデータ(1:ユーザ定義領域内、0:ユーザ定義領域外) (ビット0:エリア1 ～ビット31:エリア32)	R	整数型	341
57	M_In	入力番号(0 ～ 32767)	外部入力信号(ビット単位)を取り込みます。 汎用ビットデバイス:ビット信号入力 0=off 1=on	R	整数型	315
58	M_Inb/ M_In8	入力番号(0 ～ 32767)	外部入力信号(8ビット単位)を取り込みます。 汎用　〃　:バイト信号入力	R	整数型	315
59	M_Inw/ M_In16	入力番号(0 ～ 32767)	外部入力信号(16ビット単位)を取り込みます。 汎用　〃　:ワード信号入力	R	整数型	315
60	M_In32	入力番号(0 ～ 32767)	外部入力信号(32ビット単位)を数値で取り込みます。 汎用　〃　:ダブルワード信号入力	R	整数型	317
61	M_Out	出力番号(0 ～ 32767)	外部出力信号(ビット単位)を出します。 汎用ビットデバイス:ビット信号出力 0=off 1=on	RW	整数型	325
62	M_Outb/ M_Out8	出力番号(0 ～ 32767)	外部出力信号(8ビット単位)を出します。 汎用　〃　:バイト信号出力	RW	整数型	325
63	M_Outw/ M_Out16	出力番号(0 ～ 32767)	外部出力信号(16ビット単位)を出します。 汎用　〃　:ワード信号出力	RW	整数型	325
64	M_Out32	出力番号(0 ～ 32767)	外部出力信号(32ビット単位)に数値を出します。 汎用　〃　:ダブルワード信号出力	RW	整数型	327
65	M_DIn	入力番号(6000 ～)	CR7xx-Qシリーズでは使用できません。 (CC-Linkのリモートレジスタ　:入力レジスタ読み出し)	R	整数型	307
66	M_DOut	出力番号(6000 ～)	CR7xx-Qシリーズでは使用できません。 (CC-Linkのリモートレジスタ　:出力レジスタ書き込み、読み出し)	RW	整数型	307
67	M_HndCq	入力番号(1 ～ 8)	ハンドチェック入力信号を返します。	R	整数型	314
68	P_Safe	メカ番号(1 ～ 3)	待避点位置を返します。	R	位置型	362
69	J_Origin	メカ番号(1 ～ 3)	原点設定したときの関節座標データを返します。	R	関節型	298
70	M_Open	ファイル番号(1 ～ 8)	指定されたファイルまたは通信回線のオープン状態を返します。	R	整数型	324
71	C_Mecha	メカ番号(1 ～ 3)	ロボットの形名を返します。	R	文字列型	291
72	C_Maker	なし	メーカー側の情報を示します。(文字列64文字まで)	R	文字列型	290
73	C_User	なし	パラメータ"USERMSG"の内容を返します。(文字列 64文字まで)	R	文字列型	292
74	C_Date	なし	現在の日付を、"年/月/日"の形式で示す	R	文字列型	290
75	C_Time	なし	現在の時間を、"時間:分:秒"の形式で示します。	R	文字列型	292
76	M_BTime	なし	バッテリ残容量時間を返します。(時間)	R	整数型、時間	301
77	M_Timer	タイマー数(1 ～ 8)	常にカウントしています。値の設定もできます。[ms] プログラムで使用することによって正確なタクト時間を測定できます。	RW	単精度実数型	338

No	変数名称	配列要素数注 1)	内容 .	属性 注 2)	データの型、単位	ページ
78	P_Zero	なし	位置変数の要素(X,Y,Z,A,B,C,FL1,FL2)が全て0になっている変数	R	位置型	366
79	M_PI	なし	円周率(3.1415...)	R	倍精度実数型	329
80	M_Exp	なし	自然対数の底(2.71828...)	R	倍精度実数型	313
81	M_G	なし	重力定数(9.80665)	R	倍精度実数型	314
82	M_On	なし	常に1が設定されています。	R	整数型	323
83	M_Off	なし	常に0が設定されています。	R	整数型	323
84	M_Mode	なし	操作ﾊﾟﾈﾙのｷｰｽｲｯﾁの状態 MANUAL/AUTOMATIC(操作パネル)/AUTOMATIC(外部機器)=(1/2/3)	R	整数型	323
85	M_SplPno	メカ番号(1 ～ 3)	スプライン補間の動作において、最も新しい通過済みの経路点番号を返します。	R	整数型	335
86	M_SplVar	メカ番号(1 ～ 3)	スプライン補間の動作において、経路点に設定された任意の数値を返します。また、値が再設定されるまでの間、任意の値を保持することもできます。	RW	整数型	336
87	P_WkCord	ワーク座標番号 (1 ～ 8)	現在設定されているワーク座標の読出し、ワーク座標の設定	RW	位置型	365
88	M_ESpd	メカ番号(1 ～ 3)	Ex-T制御/Ex-Tスプライン補間の動作において、現在のEx-T座標の通過速度を返します。	R	単精度実数型、mm/s	312
89	P_ECord	メカ番号(1 ～ 3)	Ex-T制御/Ex-Tスプライン補間の動作において、使用しているEx-T座標系原点の値を返します。	R	位置型	360

注 1) メカ番号.....1 ～ 3、マルチタスク処理機能に対応したメカ番号を指定します。
スロット番号.....1 ～ 32、マルチタスク機能に対応したスロット番号を指定します。
入力番号.....0 ～ 32767(理論値)、入力信号のビット番号を指定します。
出力番号.....0 ～ 32767(理論値)、出力信号のビット番号を指定します。
注 2) R.....読み出しのみ可能です。
RW.....読み出し、書き込みとも可能です。

4.6.1 論理数

比較、入出力の結果などを表します。
整数で評価し、0 でなければ真、0 で偽を表します。代入時は、真の場合は 1 を与えます。表 4-10 に論理数を使用する処理を示します。

表 4-10：論理数の真と偽に対応する値

論理数の 1 で表されるもの	論理数の 0 で表されるもの
・比較演算の結果（真の場合） ・論理演算の結果（真の場合） ・スイッチ ON ・入出力信号 ON ・ハンド開（ハンドに電流を流す） ・割り込みなどの許可 / 有効の設定	・比較演算の結果（偽の場合） ・論理演算の結果（偽の場合） ・スイッチ OFF ・入出力信号 OFF ・ハンド閉（ハンドに電流を流さない） ・割り込みなどの禁止 / 無効の設定

4.7 関数

関数とは、与えられた引数に対して、ある決まった演算をおこないその結果を数値型または文字列型で返すもので、あらかじめ組み込まれている組込み関数と、ユーザで定義するユーザ定義関数があります。

(1) ユーザ定義関数

Def FN 文で定義します。

例）Def FNMADD(MA, MB)=MA+MB
・・・・・2 つの値の合計を求める関数 FNMADD を定義します。

関数名は FN で始まり、3 文字めにデータの型を示す識別文字（C：文字列、M：数値、P：位置、J：関節）を記述し、8 文字以内で指定します。

(2) 組込み関数

組込み関数の一覧を表 4-11 に示します。

表 4-11：組込み関数一覧

区分	関数名（書式）	機能	ページ	結果
数値関数	Abs (< 数式 >)	絶対値を与える	368	数値
	CInt (< 数式 >)	数式の小数部を四捨五入し、整数に変換する	375	
	Deg (< 数式 : ﾗｼﾞｱﾝ >)	角度単位 ﾗｼﾞｱﾝ (rad) を度 (deg) に変換する	378	
	Exp (< 数式 >)	数式の指数関数の値を与える	379	
	Fix (< 数式 >)	整数部を与える	380	
	Int (< 数式 >)	数式の値を超えない最大の整数を与える	382	
	Len (< 文字列式 >)	文字列の長さを与える	384	
	Ln (< 数式 >)	自然対数を与える	385	
	Log (< 数式 >)	常用対数を与える	385	
	Max (< 数式 >...)	任意の個数の引数の中から最大値を求める	386	
	Min (< 数式 >...)	任意の個数の引数の中から最小値を求める	387	
	Rad (< 数式 : 度 >)	角度単位 度 (deg) をラジアン (rad) に変換する	391	
	Sgn (< 数式 >)	数式の符号を調べる	399	
	Sqr (< 数式 >)	平方根を与える	403	
	Strpos(< 文字列式 >, < 文字列式 >)	第 1 引数文字列内で第 2 引数文字列の位置を求める	403	
	Rnd (< 数式 >)	乱数を発生させる	394	
	Asc(< 文字列式 >)	文字列式の最初の文字のキャラクターコードを与える	371	
	Cvi(< 文字列式 >)	2 文字の文字列式を整数に変換する	377	
	Cvs(< 文字列式 >)	4 文字の文字列式を単精度実数に変換する	377	
	Cvd(< 文字列式 >)	8 文字の文字列式を倍精度実数に変換する	378	
	Val(< 文字列式 >)	文字列を数値に変換する	404	
	SplSpd(< スプライン番号 >)	任意のスプラインファイルに登録されている経路点情報から MvSpl、EMvSpl 命令にてエラーとならず指定可能な最大速度を求める。		

区分	関数名（書式）	機能	ページ	結果
三角関数	ACos(< 数式 >)	逆余弦（アークコサイン）を与える　単位：ラジアン 定義域：-1.0 ～ +1.0　値域：0 ～ π	369	数値
	ASin(< 数式 >)	逆正弦（アークサイン）を与える　単位：ラジアン 定義域：-1.0 ～ +1.0　値域：－π/2 ～ +π/2	372	
	Atn(< 数式 >)	逆正接（アークタンジェント）を与える 単位：ラジアン 定義域：数値　値域：－π/2 ～ +π/2	372	
	Atn2(< 数式 >,< 数式 >)	逆正接（アークタンジェント）を与える 単位：ラジアン Θ=ATN2(⊿y, ⊿x) 定義域：⊿y, ⊿x のいずれかが 0 でない数値 値域：－π ～ +π	372	
	Cos(< 数式 >)	余弦（コサイン）を与える　単位：ラジアン 定義域：数値の範囲　値域：-1 ～ +1	376	
	Sin(< 数式 >)	正弦（サイン）を与える　単位：ラジアン 定義域：数値の範囲　値域：-1 ～ +1	400	
	Tan(< 数式 >)	正弦（タンジェント）を与える　単位：ラジアン 定義域：数値の範囲　値域：数値の範囲	404	
文字列関数	Bin$(< 数式 >)	数式の値を 2 進文字列に変換する	373	文字列
	Chr$(< 数式 >)	数式の値のキャラクタコードを持つ文字を与える	375	
	Hex$(< 数式 >)	数式の値を 16 進文字列に変換する	382	
	Left$(< 文字列式 >,< 数式 >)	第 1 引数の文字列の左から、第 2 引数で指定された長さの文字列を求める	384	
	Mid$(< 文字列式 >,< 数式 >,< 数式 >)	第 1 引数の文字列の、第 2 引数で指定された位置から第 3 引数で指定された長さの文字列を求める	386	
	Mirror$(< 文字列式 >)	文字列の 2 進数のビットのミラー反転を行う	387	
	Mki$(< 数式 >)	数式の値を 2 文字の文字列に変換する	388	
	Mks$(< 数式 >)	数式の値を 4 文字の文字列に変換する	388	
	Mkd$(< 数式 >)	数式の値を 8 文字の文字列に変換する	389	
	Right$(< 文字列式 >,< 数式 >)	第 1 引数の文字列の右から、第 2 引数で指定された長さの文字列を求める	394	
	Str$(< 数式 >)	数式の値を 10 進文字列に変換する	403	
	CkSum(< 文字列式 >,< 数式 >,< 数式 >)	文字列のチェックサムをつくる 第 1 引数の文字列にて第 2 引数の位置から第 3 引数の位置までの文字を加算した結果の下位バイトの値を返す	376	数値
位置変数	Dist(< 位置 >,< 位置 >)	2 点間の距離を求める	379	位置
	Fram(< 位置 1>,< 位置 2>,< 位置 3>)	3 点で指定された座標系を計算する。位置 1 は平面の原点、位置 2 は +X 軸上の点、位置 3 は +Y 軸方向の平面上の点となり 3 つの XYZ の座標から平面の原点位置と姿勢をリターン値（位置）で返す。メカ構造に無関係に 6 軸三次元で演算する。 5 軸ロボットでは ABC の姿勢データの意味が異なるため使用できません。	381	
	Rdfl1(< 位置 >,< 数値 >)	指定した位置の構造フラグを文字データで返す。 引数の < 数値 > に 0=R/L,1=A/B,2=F/N がリターンになる	392	文字
	Setfl1(< 位置 >,< 文字 >)	指定した位置の構造フラグを変更する。変更するデータは文字で指定する。(R/L/A/B/F/N)	395	
	Rdfl2(< 位置 >,< 数値 >)	指定した位置の多回転データを数値 (-2 ～ 1) で返す。 引数の < 数値 > は軸番号 (1 ～ 8) を返す。	393	数値
	Setfl2(< 位置 >,< 数値 >,< 数値 >)	指定した位置の多回転データを数値 (-2 ～ 1) で変更する 式は左辺が変更する軸番号、右辺が設定する数値。	396	
	Align(< 位置 >)	位置 1 の姿勢軸 (A,B,C) が最も近い直交姿勢 (0, ±90, ±180) をとる値を返す。 5 軸ロボットでは ABC の姿勢データの意味が異なるため使用できません。	370	
	Inv(< 位置 >)	逆行列を求める。	383	位置
	PtoJ(< 位置 >)	位置データを関節データに変換する。	391	関節
	JtoP(< 位置 >)	関節データを位置データに変換する。	383	位置

区分	関数名 (書式)	機能	ページ	結果
位置変数	Zone(< 位置 1>,< 位置 2>,< 位置 3>)	位置 1 が、位置 2 と位置 3 の 2 点で構成されている空間 (立方体) に入っているかどうかのチェック 範囲外 =0, 範囲内 =1 チェックしない、または、存在しない位置成分については、単位が deg の場合、位置 2 に -360、位置 3 に 360 単位が mm の場合、位置 2 に -10000、位置 3 に 10000 を対応する成分に設定する。	405	数値
	Zone2(< 位置 1>,< 位置 2>,< 位置 3> 数値 1, 数値 2, 数値 3, 位置 4)	位置 1 が、位置 2 と位置 3 の 2 点で構成されている空間 (円筒体) に入っているかどうかのチェック 範囲外 =0, 範囲内 =1 対象は XYZ の座標値のみ ABC の姿勢データは無視。	406	数値
	Zone3(< 位置 1>,< 位置 2>,< 位置 3>, < 位置 4> 数値 1, 数値 2, 数値 3)	位置 1 が、位置 2 と位置 3 と位置 4 の 3 点と数値 1、数値 2、数値 3 で構成されている空間 (立方体) に入っているかどうかのチェック 範囲外 =0, 範囲内 =1 対象は XYZ の座標値のみ ABC の姿勢データは無視。	406	数値
	PosCq(< 位置 >)	< 位置 > が動作範囲内かをチェックします。	390	数値
	PosMid(< 位置 1>,< 位置 2>, 数値 1, 数値 2)	< 位置 1> と < 位置 2> の中間位置を演算します。	390	位置
	CalArc(< 位置 1>,< 位置 2>,< 位置 3> 数値 1, 数値 2, 数値 3, 位置 4)	< 位置 1>,< 位置 2>,< 位置 3> よりできる円弧の情報を出力します。	374	数値
	SetJnt(<J1 軸 >,<J2 軸 >,<J3 軸 >,<J4 軸 > <J5 軸 >,<J6 軸 >,<J7 軸 >,<J8 軸 >)	関節変数へ値を設定します。	397	関節
	SetPos(<X 軸 >,<Y 軸 >,<Z 軸 >,<A 軸 > <B 軸 >,<C 軸 >,<L1 軸 >,<L2 軸 >)	位置変数へ値を設定します。	398	位置
	SplPos(< スプライン番号 >, < 経路点番号 >,< フレーム変換 >)	任意のスプラインファイルに登録されている経路点データを取り出します。	401	位置
	SplECord(< スプライン番号 >, < フレーム変換 >)	任意のスプラインファイルに登録されている Ex-T 座標系原点データを取り出します。	400	位置

4.8 命令語の一覧

以下に、各命令語の説明頁一覧を示します。使用頻度の高いものと思われる順で表示してあります。

(1) 動作制御関連の命令

命令	解説	ページ
Mov (ムーブ)	関節補間	233
Mvs (ムーブエス)	直線補間	243
Mvr (ムーブアール)	円弧補間	237
Mvr2 (ムーブアール 2)	円弧補間 2	239
Mvr3 (ムーブアール 3)	円弧補間 3	241
Mvc (ムーブシー)	円補間	236
Mva (ムーブアーチ)	アーチモーション補間	234
MvSpl (ムーブスプライン)	スプライン補間	245
Mxt (ムーブエクスターナル)	外部からの位置指令による直接動作	250
MvTune (ムーブチューン)	用途に応じた最適動作モード指定	248
Ovrd (オーバーライド)	全体の速度指定	256
Spd (スピード)	直線、円弧補間動作時の速度指定	269
JOvrd (ジェーオーバーライド)	関節補間動作時の速度指定	229
Cnt (コント)	連続軌跡モード指定	185
Accel (アクセル)	加減速度指定	165
ColChk (コルチェック)	衝突検知	187

命令	解説	ページ
CavChk On (キャブチェック オン)	干渉回避	174
Cmp Jnt (コンプジョイント)	関節座標系でのコンプライアンス指定	177
Cmp Pos (コンプポーズ)	直交座標系でのコンプライアンス指定	179
Cmp Tool (コンプツール)	ツール座標系でのコンプライアンス指定	181
Cmp Off (コンプオフ)	コンプライアンス設定無効	183
CmpG (コンプゲイン)	コンプライアンスのゲイン指定	184
Oadl (オーエーディーエル)	最適加減速指定	251
LoadSet (ロードセット)	ハンドの付加条件指定	232
Prec (プレック)	高精度モード指定	258
Torq (トルク)	各軸のトルク指定	277
JRC (ジョイント・ロール・チェンジ)	先端軸を多回転可能	230
Fine (ファイン)	ロボットの位置決め範囲指定 (パルス)	217
Fine J (ファイン ジョイント)	ロボットの位置決め範囲指定 (関節軸)	218
Fine P (ファイン ポーズ)	ロボットの位置決め範囲指定 (直線距離)	219
Servo (サーボ)	サーボモータ電源の ON/OFF	266
SpdOpt (スピードオプティマイズ)	直線補間動作時の速度調整	270
Wth (ウィズ)	動作命令への付随命令	280
WthIf (ウィズイフ)	動作命令への条件付き付随命令	280
EMvs (イームーブエス)	Ex-T 制御直線補間	210
EMvc (イームーブシー)	Ex-T 制御円補間	204
EMvr (イームーブアール)	Ex-T 制御円弧補間	205
EMvr2 (イームーブアール 2)	Ex-T 制御円弧補間 2	207
EMvr3 (イームーブアール 3)	Ex-T 制御円弧補間 3	208
EMvSpl (イームーブスプライン)	Ex-T スプライン補間	212

(2) プログラム制御関連の命令

命令	解説	ページ
Rem(レム)	コメント (')	261
If…Then…Else…EndIf(イフ ゼン エルス)	条件分岐	226
Select Case (セレクト ケース)	複数の分岐が可能	265
GoTo(ゴートゥー)	ジャンプ	224
GoSub(～ Return) (ゴーサブ リターン)	サブルーチンジャンプ	223
Reset Err(リセット エラー)	エラーをリセットします (デフォルトは使用禁止)。	262
CallP (コールピー)	プログラムコール	171
FPrm (エフパラム)	プログラムコールの引数定義	221
Dly (ディレイ)	タイマ	203
Hlt (ホルト)	プログラム中断	224
End (エンド)	プログラム終了	215
On… GoSub (オン ゴーサブ)	値によりサブルーチンジャンプ	253
On…GoTo (オン ゴートゥー)	値によりジャンプ	254
For ～ Next (フォー ネクスト)	繰り返し	220
While ～ WEnd(ホワイル エンド)	条件付き繰り返し	279
Open (オープン)	ファイルを開く	255

命令	解説	ﾍﾟｰｼﾞ
Print（プリント）	データを出力	259
Input（インプット）	データを入力	228
Close（クローズ）	ファイルを閉じる	175
ColChk（コルチェック）	衝突検知機能の有効/無効	187
HOpen/HClose（ハンドオープン/ハンドクローズ）	ハンドの開閉	225
Error（エラー）	ユーザエラー	216
Skip(スキップ)	動作中のスキップ	268
Wait（ウェイト）	条件待ち	278
Clr（クリア）	信号クリア	176
SplWrt（スプラインライト）	スプラインファイル作成	273
SplFWrt（スプラインフレームライト）	スプラインファイル内フレーム変換情報書き換え	272

(3) 定義命令

命令	解説	ﾍﾟｰｼﾞ
Dim（ディム）	配列変数宣言	202
Def Plt（デフ パレット）	パレット宣言	199
Plt（パレット）	パレット位置計算	257
Def Act（デフ アクト）	割り込み定義	191
Act（アクト）	割り込み監視開始、終了	166
Def Arch（デフ アーチ）	アーチモーション動作のアーチ形状の定義	193
Def Jnt(デフ ジョイント)	関節位置変数定義	198
Def Pos（デフ ポーズ）	直交位置変数定義	201
Def Float/Def Double/Def Inte/Def Long（ﾃﾞﾌ ﾌﾛｰﾄ/ﾀﾞﾌﾞﾙ/ｲﾝﾃｼﾞｬｰ/ﾛﾝｸﾞ）	整数、実数変数定義	196
Def Char（デフ キャラ）	文字変数の定義	194
Def IO（デフアイオー）	信号変数の定義	197
Def FN（デフ ファンクション）	ユーザ関数の定義	195
Title（タイトル）	プログラムタイトルの設定	275
Base（ベース）	ロボットベース位置の設定	168
Tool（ツール）	ツール長の設定	276
SetCalFrm（セット キャリブレーションフレーム）	フレーム変換で使用する座標系の設定	267

(4) マルチタスク関連

命令	解説	ﾍﾟｰｼﾞ
XLoad（エックスロード）	別タスクスロットへのプログラムロード	282
XRun（エックスラン）	別タスクスロットのプログラムを実行	284
XStp（エックスストップ）	別タスクスロットのプログラムを停止	285
XRst（エックスリセット）	中断中の別タスクスロットをプログラムリセット	283
XClr（エックスクリア）	指定したタスクスロットからプログラムロードを解除	281
GetM（ゲットメカ）	メカ制御権の取得	222
RelM（リリースメカ）	メカ制御権の解放	261
Priority（プライオリティ）	タスクスロットの優先度を変更	260
Reset Err(リセット エラー)	エラーのリセット（デフォルトは使用禁止）	262

(5) その他

命令	解説	ﾍﾟｰｼﾞ
ChrSrch（キャラクタ サーチ）	文字配列の中から文字列を検索します。	173
GetPos(ゲットポジション)	予約	-

4.9 演算子

数値の実数または整数の型宣言はおこないません。そのかわりに、演算の種類によって強制的に型が変換されます。(表 4-12 参照。)左項と右項のデータ型の組み合せによる演算結果の型は以下の通りです。

例）

	左項	演算	右項	演算結果
・	15	And	256	15
	(数値型)		(数値型)	(数値型)
・	P1	*	M1	P2
	(位置型)		(数値型)	(位置型)
・	M1	*	P1	
	(数値型)		(位置型)	記述エラー

表 4-12：演算によるデータ型変換表

左項型	演算	右項型				
		文字列	数 値		位置	関節
			整数	実数		
文字列	代入 =	文字列	−	−	−	−
	加算 +	文字列	−	−	−	−
	比較(比較演算子)	整数	−	−	−	−
整 数	加算 +	−	整数	実数	−	−
	減算 −	−	整数	実数	−	−
	乗算 *	−	整数	実数	−	−
	除算 /	−	整数	実数	−	−
	整数除算 ¥	−	整数	整数	−	−
	剰余 Mod	−	整数	整数	−	−
	指数 ^	−	整数	実数	−	−
	代入 =	−	整数	整数	−	−
	比較(比較演算子)	−	整数	整数	−	−
	論理(論理演算子)	−	整数	整数	−	−
実 数	加算 +	−	実数	実数	−	−
	減算 −	−	実数	実数	−	−
	乗算 *	−	実数	実数	−	−
	除算 /	−	実数	実数	−	−
	整数除算 ¥	−	整数	整数	−	−
	剰余 Mod	−	整数	整数	−	−
	指数 ^	−	実数	実数	−	−
	代入 =	−	整数	実数	−	−
	比較(比較演算子)	−	整数	整数	−	−
	論理(論理演算子)	−	整数	整数	−	−
位 置	加算 +	−	−	−	位置	−
	減算 −	−	−	−	位置	−
	乗算 *	−	位置	位置	位置	−
	除算 /	−	位置	位置	位置	−
	整数除算 ¥	−	−	−	−	−
	剰余 Mod	−	−	−	−	−
	指数 ^	−	−	−	−	−
	代入 =	−	−	−	位置	−
	比較(比較演算子)	−	−	−	−	−
	論理(論理演算子)	−	−	−	−	−
関 節	加算 +	−	−	−	−	関節
	減算 −	−	−	−	−	関節
	乗算 *	−	関節	関節	−	−
	除算 /	−	関節	関節	−	−
	整数除算 ¥	−	−	−	−	−
	剰余 Mod	−	−	−	−	−
	指数 ^	−	−	−	−	−
	代入 =	−	−	−	−	関節
	比較(比較演算子)	−	−	−	−	−
	論理(論理演算子)	−	−	−	−	−
右項のみ(単項)	反転 −	−	整数	整数	位置	関節
	否定 Not	−	整数	整数	−	−

反転：符号反転　　否定：論理否定　　代入：代入演算　　剰余：剰余演算
比較：比較演算　　論理：論理演算(論理否定を除く)

［注意］

・’－’を記述した部分の演算は、定義されていません。
・整数と整数の乗除算の結果は、乗算の場合は整数型、除算の場合は実数型となります。
・右項が 0 の除算の場合（0 で割る場合）、演算できません。
・剰余演算、論理演算（否定を含む）においては、実行時、実数をすべて強制的に 整数に変換（四捨五入）して演算します。

4.10 演算の優先順位

式の中に複数の演算子が含まれていた場合は、以下の表 4-13 に示す優先順位で演算が行われます。

表 4-13：演算の優先順位

演算（演算子）	演算の種類	優先順位
1) カッコ () 内の演算		高
2) 関数	関数	:
3) 指数（^）	数値演算	:
4) 単項演算子 (+,-)	数値演算	:
5) ＊ ／	数値演算	:
6) ￥	数値演算	:
7)Mod	数値演算	:
8) ＋ －	数値演算	:
9)《 》	論理演算	:
10) 比較演算子	比較演算	:
(=,<>,>,<,<=,=<,>=,=>)		:
11)Not	論理演算	:
12)And	論理演算	:
13)Or	論理演算	:
14)Xor	論理演算	低

4.11 プログラムの制御構造の深さ

プログラムを組む場合、制御構造の深さについて考慮する必要があります。
表 4-14 に示したコマンドを用いるとプログラムの制御構造が一段深くなります。各コマンドの制御構造の深さには制限があり、これを超えると実行時にエラーになります。

表 4-14：制御構造の深さ制限

	段数	処理対応コマンド
プログラム内ユーザスタック	16 段	・繰り返し制御（For ～ Next, While ～ WEnd）
	8 段	・関数呼び出し（CallP）
	800 段（最大）	・サブルーチン呼び出し（GoSub） For ～ Next, While ～ WEnd, Call 命令の使用頻度によって減少します。

4.12 予約語

予約語とはシステムで既に使用されている語のことです。プログラム上で予約語と同一の名称は使用できません。
命令、関数、システム状態変数（ロボット状態変数）などが予約語となります。

4.13 命令の詳細説明

4.13.1 記載項目の見方

【機能】　　：命令語の機能を示します。
【書式】　　：命令語の引数の入力方法を示します。
　　<>は引数を示します。
　　[]は省略可能を示します。
　　□ はスペースが必要なことを示します。
【用語】　　：引数の意味、範囲などを示します。
【文例】　　：文例を示します。
【解説】　　：詳細な機能や、注意事項を示します。
【使用可能なロボット機種】：使用できるロボットの機種を示します。
【関連パラメータ】　：関連するパラメータを示します。
【関連システム状態変数】：関連するシステム変数を示します。
【関連命令】　　：関連する命令語を示します。

4.13.2 各命令語の説明

以下にアルファベット順に各命令の解説を示します。

Accel（アクセル）

【機能】

ロボット動作時の加速度および減速度を割合(%)で指定します。
最適加減速時にも有効です。

※ 最適加減速度時の加減速時間は、Oadl 命令使用時に最適に算出される時間で、M_SetAdl 変数の値が考慮されます。

【書式】

Accel □ [< 加速度割合(%)>],[< 減速度割合(%)>]
,[<Mva 命令上昇時加速度割合(%)>],[<Mva 命令上昇時減速度割合(%)>]
,[<Mva 命令下降時加速度割合(%)>],[<Mva 命令下降時減速度割合(%)>]

【用語】

<加速･減速割合>	1～100(%)。速度0から最高速度に達するまで の加減速度を割合で指定します。定数または変数で記述することができます。省略時は100,100で最高加減速度になります。単位:[%]
<上昇時加減速度割合>	Mva命令でのアーチモーション動作の上昇時の加減速割合を指定します。省略時は100です。定数または変数で記述することができます。
<下降時加減速度割合>	Mva命令でのアーチモーション動作の下降時の加減速割合を指定します。省略時は100です。定数または変数で記述することができます。

【文例】

```
1 Accel 50,100                    ' 重負荷指定（標準加減速時間が 0.2 秒の場合、加速時間 0.4、減速時間 0.2 秒になります）
2 Mov P1
3 Accel 100,100                   ' 標準負荷指定
4 Mov P2
5 Def Arch 1,10,10,25,25,1,0,0
6 Accel 100,100,20,20,20,20       ' Mva 命令動作時の上昇、下降時のオーバーライドを 20 に設定
7 Mva P3,1
```

【解説】

(1) 標準加減速時間はロボット毎に決められており、それに対する割合(%)を設定します。
システム初期値は 100%、100% です。（最高加減速度）
(2) 本命令を実行して変更した加速割合は、プログラムリセット時、および End 文実行時にシステム初期値に再設定されます。
(3) 100% 以上記述することは可能ですが、機種によっては内部的に上限を 100% にしている機種もあります。100% を越えて設定した場合はロボット本体の寿命に影響を与える可能性があります。また、速度オーバーエラーや過負荷エラーが発生しやすくなりますので 100% 以上に設定する場合は十分ご注意願います。
(4) Cnt 有効時の円滑動作は、加速度や動作速度によって軌跡経路が異なります。また、一定速度で円滑動作させる場合は、加速度と減速度を同一値にしてください。初期状態では Cnt 無効となっています。
(5) 最適加減速制御時(Oadl　On)でも有効になります。

【関連命令】

Oadl（オーエーディーエル）、LoadSet（ロードセット）

【関連システム状態変数】

M_Acl/M_DAcl/M_NAcl/M_NDAcl/M_AclSts

【関連パラメータ】

JADL

Act（アクト）

【機能】

動作中の信号などによる割り込み処理の許可 / 禁止を指定します。

【書式】

Act □ < 優先番号 >=<1/0/-1>

【用語】

〈優先番号〉　0:割り込み全体を許可または禁止します。
1～8:Def Act文で定義した割り込みの優先番号を指定します。
優先番号は､Act命令の後に必ずスペースを空けて記述してください。
Act1と続けて記述すると､変数名の宣言文となります。

〈1/0/-1〉　1:割り込みを許可する場合
0:禁止にする場合、変数も可。
-1:〈優先番号〉が1～8の場合､割り込み処理中を解除します｡なお､割り込み監視も停止します。

【文例】

①P1 から P2 へ移動中、入力信号 1 がオンしたら（1 になったら）その信号が 0 になるまでループする

```
 1 Def Act 1,M_In(1)=1 GoSub *INTR       ' 割り込み 1 の条件に入力信号 1 を割り当てます。
 2 Mov P1
 3 Act 1=1                               ' 割り込み 1 を許可します。
 4 Mov P2
 5 Act 1=0                               ' 割り込み 1 を禁止にします。
    :
10 *INTR                                 '
11 If M_In(1)=1 GoTo *INTR               ' M_IN(1) 信号が 0 になるまでループ
12 Return 0                              '
```

②P1 から P2 へ移動中、入力信号 1 がオンしたら（1 になったら）動作を中断し出力信号 10 をオンします

```
1 Def Act 1,M_In(1)=1 GoSub *INTR        ' 割り込み 1 の条件に入力信号 1 を割り当てます。
2 Mov P1
3 Act 1=1                                ' 割り込み 1 を許可します。
4 Mov P2
   :
10 *INTR
11 Act 1=0                               ' 割り込み 1 を禁止にします。
12 M_Out(10)=1                           ' 出力信号 10 をオンする。
13 Return 1                              ' 割り込みが発生した次の行へ戻ります。
```

③P1 から P2 へ移動中、入力信号 1 がオンしたら（1 になったら）動作を中断し、信号が 0 になるまで待った後、最初に戻ります

```
1 *Start
   :
11 Def Act1, M_In(1)=1 Goto *INTR2       ' 割り込み 1 の条件に入力信号 1 を割り当てます。
12 Mpv P1
13 Act 1=1                               ' 割り込み 1 を許可します。
14 Mov P1
   :
21 *INTR2
22 Act 1=0                               ' 割り込み 1 を禁止にします。
23 *Loop
24 If M_In(1) <> 0 Then Goto *Loop
25 Act 1=-1                              ' 割り込み処理中を解除します。なお、割り込み監視も停
                                           止します
26 Goto *Start
```

【解説】
(1) 優先番号 0 は、プログラム運転開始時は " 許可 " の状態です。 < 優先番号 >0 を禁止に設定すると優先番号 1 ～ 8 が " 許可 " になっていても割り込みは行われません。
(2) 優先番号 1 ～ 8 は、プログラム運転開始時は、" 禁止 " の状態です。
(3) 割り込みは、以下の条件が成立しているときに行われます。
・< 優先番号 >0 が " 許可 "
・Def Act 文の条件が成立
・Def Act で指定した < 優先番号 > を Act 文で有効にしている時
(4) 割り込み処理からのリターンは Return 0 または Return 1 のどちらかで記述し、割り込み処理終了後の戻り先を指定します。但し、割り込み処理から Return 1 で次の行へ戻る場合は、割り込み処理内で割り込み禁止にしてください。割り込み禁止にせず、割り込み条件が成立したままの場合、再度割り込み処理を実行し次の行へ戻るため、行が実行されずスキップすることがあります。
(5) ロボットが補間動作中の場合でも中断して、Def Act 文で定義した割り込み処理が実行されます。
(6) 割り込み処理中には、その < 優先番号 > は " 禁止 " 状態で実行されます。
Return 命令で割り込み処理から戻ると再び " 許可 " 状態に戻ります。
(7) 通信割り込み (Com) は Def Act 文で定義した割り込みより優先順位が高くなります。
(8) 他の割り込みとの優先順位関係は以下の通りです。
Com > Act > WthIf(Wth)
(9) Act 0=-1 を指定した場合は、L3110 エラーが発生します。
(10) -1 の指定は、バージョンが R5d/S5d 以降で使用できます。

【関連命令】
Def Act (デフ アクト)、Return (リターン)

Base（ベース）

【機能】

ロボットの現在位置制御の基準であるワールド座標系を変更（移動、および回転）することができます。ベース変換データを直接指定する方法と、予め定義したワーク座標系の番号を指定する方法があります。本機能はロボットの教示データや、ジョグ動作へ大きく影響します。「4.5 ロボットの座標系解説」をご一読の上，十分注意してご使用願います。

【書式】

```
Base □＜ベース変換データ＞        ’ベース変換データを直接指定
Base □＜ベース座標番号＞          ’ベース座標番号（ワーク座標系番号）でベース変換データを間接的に指定
```

【用語】

〈ベース変換データ〉　位置定数または位置変数でベース変換データを指定します。
設定する値(座標値)は､新しく設定するワールド座標系から見たベース座標系原点の位置データです。

〈ベース座標番号〉　システム初期値､またはパラメータ(ワーク座標系)に設定されている値をベース変換データとして指定します。
数値型の定数､または変数で､値0～8の範囲で指定します。

0:　P_NBase(システム初期値)を指定(P_NBase=0,0,0,0,0,0のためベース変換をクリアします)

1～8:　パラメータ ワーク座標系:WK1CORD～WK8CORDに対応

注)実数､または倍精度実数で指定した場合は小数点以下の値を切り捨てます。

【文例】

ベース変換データで指定

```
1 Base (50,100,0,0,0,90)        ’新しいワールド座標系を定数による変換データで設定
2 Mvs P1                        ’新しいワールド座標系で P1 へ移動
3 Base P2                       ’新しいワールド座標系を変数による変換データで設定
4 Mvs P1                        ’新しいワールド座標系で P1 へ移動
5 Base 0                        ’ワールド座標系を初期値に戻す。
                                  (P_NBase（システム初期値）をベース変換データに設定）
```

ベース座標番号で指定

```
1 Base 1                        ’ワーク座標系 1（パラメータ：WK1CORD）を新たにワールド座標系とする
2 Mvs P1                        ’新しいワールド座標系で P1 へ移動
3 Base 2                        ’ワーク座標系 2（パラメータ：WK2CORD）を新たにワールド座標系とする
4 Mvs P1                        ’新しいワールド座標系で P1 へ移動
5 Base 0                        ’ワールド座標系を初期値に戻す。
                                  (P_NBase（システム初期値）をベース変換データに設定）
```

【解説】

(1) ベース変換する値 (座標値) は、新しく設定するワールド座標系から見たベース座標系原点の位置データです。
このため、ロボットの現在位置を使用して Fram 関数などで定義した座標値をベース変換データに指定する場合は、座標値を逆変換して設定します。(例 : Base Inv(P1))
なお、ワーク座標系番号を指定する場合は、上記逆変換処理を内部で自動的におこないます。
位置データの X，Y，Z 成分は、ワールド座標系原点からベース座標系原点までの平行移動量を表します。また、A，B，C 成分は、ベース座標系がロボット座標系に対してどれだけ傾いているかを表します。

X・・・X 軸方向へ平行移動する距離
Y・・・Y 軸方向へ平行移動する距離
Z・・・Z 軸方向へ平行移動する距離
A・・・X 軸回りに回転させる角度
B・・・Y 軸回りに回転させる角度
C・・・Z 軸回りに回転させる角度

A，B，C はワールド座標系の原点から正方向にみて、時計回りを正の回転方向にとります。

(2) 構造フラグの内容は意味をもちません。

(3) 本命令で変更したベース座標系は、パラメータ MEXBS に保存され、コントローラの電源 OFF 後も保持されます。

(4) ベース変換をおこなうと、ロボットの現在位置は新しく設定されたワールド座標系またはワーク座標系を基準とした値に変わります。移動命令で指定された目的位置も、新しく設定されたワールド座標系またはワーク座標系内の位置として扱われます。

(5) ベース変換データは、ロボットの機種 (本体の構造) によってその有効な軸成分が異なります。432 ページの「5.6 標準ツール座標について」を参照してください。
また，本命例に関連する説明箇所を以下に示しますので，併せて参照願います。
147 ページの「4.5.1 ロボットの座標系について」
148 ページの「4.5.2 ベース変換について」

(6) このデータのシステム初期値は P_NBase=(0,0,0,0,0,0)(0,0) です。

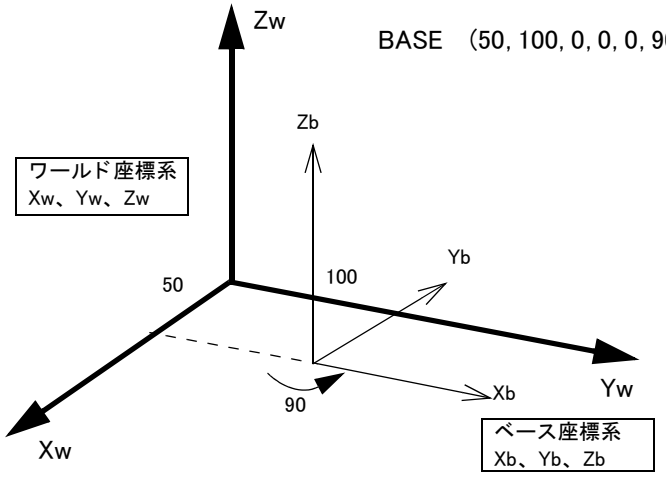


図 4-11 : 座標系の概念図

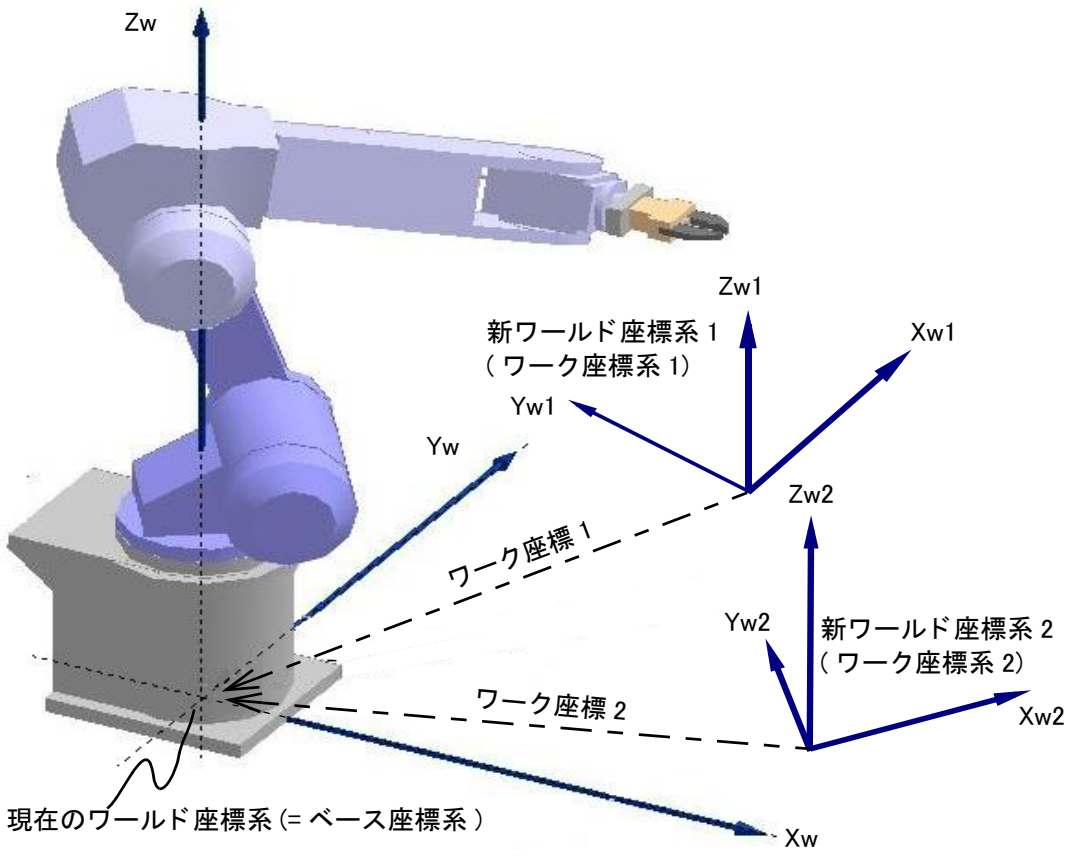


図 4-12 : ワーク座標系番号を指定したベース変換

⚠注意

ベース変換をおこなうと、ロボットの現在位置は新しく設定されたワールド座標系またはワーク座標系を基準とした値に変わります。移動命令で指定された目的位置も、新しく設定されたワールド座標系またはワーク座標系内の位置として扱われます。
したがって、それまでの教示データはそのままでは使えなくなる場合があります。教示時の座標系と新しく設定された座標系が同一である必要があります。
座標系が一致していないと、ロボットは思わぬ位置へ移動し、物損や人身事故の原因となります。
ベース変換機能をご使用になる場合は、変換するベース座標系と教示位置の関係をしっかりと管理して、ロボットを正しく動作させ、ベース変換機能を効果的にご活用いただきますようお願いします。

【関連パラメータ】
MEXBS、WKnCORD (n は 1 ～ 8)、MEXBSNO
【関連システム状態変数】
M_BsNo、P_Base/P_NBase、P_WkCord

CallP（コールピー）

【機能】

指定したプログラムを実行します。(GoSub のプログラム呼び出し版です)
サブプログラムでは End 命令または最終行にてメインプログラムに戻ります。

【書式】

CallP □"<プログラム名>" [, <引数>[, <引数>]・・・]

【用語】

<プログラム名>　文字列定数または文字列変数でプログラムを指定します。
プログラム名の基準に関しては135ページの「(1) プログラム名」を参照してください。

<引数>　プログラムが呼び出された時に、そのプログラムに引き渡される変数または定数を指定します。引数の最大個数は16です。

【文例】

① 呼び出すプログラムに引数を渡すとき

メインプログラム

```
 1 M1=0
 2 CallP "10" ,M1,P1,P2
 3 M1=1
 4 CallP "10" ,M1,P1,P2
     :
10 CallP "10", M2,P3,P4
     :
15 End
```

"10" サブプログラム側

```
 1 FPrm M01, P01,P02
 2 If M01<>0 Then GoTo *LBL1
 3 Mov P01
 4 *LBL1
 5 Mvs P02
 6 End              ' ここでメインプログラムに戻ります。
```

※ メインプログラムのステップ番号 2、4 が実行された時、それぞれ M1, P1, P2 がサブプログラムの M01, P01, P02 に設定され、メインプログラムのステップ番号 10 が実行されたとき、M2, P3, P4 がサブプログラムの M01, P01, P02 に設定されます。

② 呼び出すプログラムに引数を渡さないとき

メインプログラム

```
 1 Mov P1
 2 CallP "20"
 3 Mov P2
 4 CallP "20"
 5 End
```

"20" サブプログラム側

```
 1 Mov P1           ' サブプログラムの P1 はメインプログラムの P1 とは異なります。
 2 Mvs P002
 3 M_Out(17)=1
 4 End              ' ここでメインプログラムに戻ります。
```

【解説】

(1) CallP 命令で実行されるサブプログラムは End 命令にて終了し、メインプログラムに戻ります (GoSub 実行時の Return 命令に相当します)。End 命令がない場合は、最終行実行後、メインプログラムに戻ります。
(2) 引数を渡す場合、CallP 命令で引数を記述するとともに、サブプログラムの先頭にて FPrm 命令により引数の定義が必要となります。
(3) メインプログラムとサブプログラムで、引数の型および個数が異なる場合は、実行時エラーとなります。
(4) プログラムリセットをした場合は、メインプログラムの先頭に制御が戻ります。
(5) メインプログラムで実行した定義文 (Def Act, Def FN, Def Plt, Dim 命令) は、サブプログラムでは無効です。サブプログラムから戻った時点で再度有効となります。
(6) ツールデータはサブプログラムでも有効です。Accel、Spd、Ovrd、JOvrd の値は無効です。
Oadl のモードは有効です。
(7) サブプログラム中で CallP を使用して、さらに別のサブプログラムを実行することが可能です。
ただし、メインプログラムや既に別のタスクスロットで実行中のプログラムを呼び出すことはできません。また、自分自身のプログラムを呼び出すこともできません。
(8) 最初のメインプログラムから、8 段階 (階層) のサブプログラム実行 (CallP) が可能です。
(9) メインプログラムからサブプログラムへ、引数により変数の値を渡すことはできますが、サブプログラムでの処理結果を引数に代入して、メインプログラムへ渡すことはできません。サブプログラムの処理結果をメインプログラムで使用する場合は、外部変数を使用して値を渡すようにしてください。

【関連命令】

FPrm (エフパラム)

ChrSrch（キャラクタ サーチ）

【機能】

文字配列の中から文字列を検索します。

【書式】

ChrSrch □＜文字列配列変数＞,＜文字列＞,＜検索結果格納先＞

【用語】

＜文字列配列変数＞	検索対象となる文字列配列を指定します。
＜文字列＞	検索する文字列を指定します。
＜検索結果格納先＞	検索する文字列がみつかった要素番号が設定されます。

【文例】

```
1 Dim C1$(10)
2 C1$(1)="ABCDEFG"
3 C1$(2)="MELFA"
4 C1$(3)="BCDF"
5 C1$(4)="ABD"
6 C1$(5)="XYZ"
7 C1$(6)="MELFA"
8 C1$(7)="CDF"
9 C1$(8)="ROBOT"
10 C1$(9)="FFF"
11 C1$(10)="BCD"
12 ChrSrch C1$(1), "ROBOT", M1        ' M1 には 8 が設定されます。
13 ChrSrch C1$(1), "MELFA", M2        ' M2 には 2 が設定されます。
```

【解説】

(1) 文字列配列変数から文字列を検索し、完全に一致する文字列配列の要素番号が＜検索結果格納先＞に設定されます。部分的に一致するものは検索されません。
上記文例で
ChrSrch C1$(1), "ROBO", M1
と記述しても、一致する文字列は検索されません。

(2) 検索する文字列が見つからなかった場合は、＜検索結果格納先＞には 0 が設定されます。

(3) 文字列の検索は、要素番号 1 から順番におこない、最初に見つかった要素番号が設定されます。
上記文例で、
ChrSrch C1$(3), "MELFA", M2
と記述しても、M2 には 2 が設定されます。(C1$(2) と C1$(6) に同じ文字列が設定されている)

(4) 検索対象となる＜文字列配列変数＞は、一次元配列のみです。二次元以上の変数が指定された場合は、実行時にエラーとなります。

CavChk On（キャブチェック オン）

【機能】

干渉回避の停止機能を有効にします。

本機能は対応機種を限定しています。詳細については「5.24 干渉回避機能について」を参照してください。

【書式】

CavChk □ <On/Off>[,< ロボット CPU 番号 >[,NOErr]]

【用語】

<On/Off>	On：干渉回避の停止機能を有効にします。 Off：干渉回避の停止機能を無効にします。
< ロボット CPU 番号 >	干渉回避の停止機能を有効 / 無効にするロボット CPU を指定します。 省略または 0 を指定した場合は、接続されているロボットすべてを対象とします。 ０～３、定数または変数で記述します。 なお、この引数は CR7xx-Q コントローラのみ機能します。
NOErr	干渉検知時にアラームを発生させません。 省略するとアラーム (L4931) が発生します。

【文例】

500 ページの「5.24.10 サンプルプログラム」を参照。

【解説】

(1) 干渉回避の停止機能を有効にします。初期状態はパラメータ CAV の設定によります。
パラメータ CAV で使用不可に設定されている場合は、アラーム (L4930) が発生します。
(2) ロボット CPU 番号を省略した場合は、装着されているロボット CPU 間のうち、干渉回避が有効な CPU 間でチェックします。
(3) ロボット CPU 番号を指定した場合、指定した先のロボット CPU が干渉回避無効であればアラーム (L4930) が発生します。
(4) ハンド、ワークモデルは LoadSet 命令で指定が可能です。
(5) トラッキング有効中は有効にできません。(アラーム (L4936) 発生)

Close（クローズ）

【機能】
指定したファイル番号のファイル（通信回線を含む）を閉じます。

【書式】

Close □ [[#]<ファイル番号>[,　[[#]<ファイル番号>・・・]]]

【用語】
〈ファイル番号〉　1～8の閉じたいファイル番号を指定します。数値定数のみです。
省略時はオープン中の全ファイルを閉じます。

【文例】
```
1 Open "temp.txt" For Append As #1 ' temp.txt というファイルを Append モードでファイル番号 1 としてオープンします。
2 Print #1, "abc"                  ' ファイルに "abc" を書き込みます。
3 Close #1                         ' ファイルを閉じます。
```

【解説】
(1) Open 命令によって開かれていたファイルを閉じます。バッファ内にデータが残っていた場合は掃き出しをおこないます。
ファイルを閉じる時、バッファに残っているデータは以下のように処理されます。

表 4-15：ファイルを閉じた時の各バッファの処理動作

バッファの種類	ファイルを閉じた時の処理
通信回線 受信バッファ	バッファ内容は破棄されます。
通信回線 送信バッファ	(Print 命令実行で即送信されるので、送信バッファにはデータは残りません)
ファイル 読出しバッファ	バッファ内容は破棄されます。
ファイル 書込みバッファ	バッファ内容をファイルに書き込んだ後、ファイルを閉じます。

(2) End 文の実行によっても閉じられます。
(3) ファイル番号を省略した場合はすべてのファイルを閉じます。

【関連命令】
Open（オープン）、Print（プリント）、Input（インプット）

Clr（クリア）

【機能】
汎用出力信号 / プログラム内ローカル数値変数 / 数値外部変数をクリアします。

【書式】

Clr □＜タイプ＞

【用語】
〈タイプ〉　定数または変数を指定できます。
0:以下の1～3の全てを実行します。
1:汎用出力信号を出力リセットパターンに基づいてクリアします。出力リセットパターンは、パラメータ:ORS10000～ORS1816で指定します。
447ページの「5.14 出力信号リセットパターンについて」を参照してください。
(0:OFF、1:ON、*:保持)
2:プログラム内で使用しているローカルな数値変数および数値配列変数の全てを0クリアします。
3:外部数値変数および外部数値配列変数の全て(システム外部変数とユーザ定義外部変数)を0クリアします。外部位置変数はクリアしません。

【文例】
① 汎用出力信号を出力リセットパターンに基づいて出力します。

```
1 Clr 1
```

② プログラム内のローカルな数値変数、数値配列変数を0クリアします。

```
1 Dim MA(10)
2 Def Inte IVAL
3 Clr 2              'MA(1)～MA(10)、IVAL、および、ﾌﾟﾛｸﾞﾗﾑ内のﾛｰｶﾙな数値変数を0クリア
```

③ 外部数値変数および外部数値配列変数全てを0クリアします。

```
1 Clr 3
```

④ 上記①～③を同時におこないます。

```
1 Clr 0
```

【関連パラメータ】
ORS10000 ～ ORS1816

【関連システム状態変数】
M_In/M_Inb/M_In8/M_Inw/M_In16、M_Out/M_Outb/M_Out8/M_Outw/M_Out16

Cmp Jnt（コンプジョイント）

【機能】

関節座標系で軸を指定しロボットを柔らかく制御するモード(コンプライアンスモード)を開始します。
注）本命令は対応機種を限定しています。【使用可能なロボット機種】を参照ください。

【書式】

Cmp □ Jnt, ＜軸指定＞

【用語】

〈軸指定〉　柔らかく制御する軸をビットパターンで指定します。
1:有効､0:無効　&B000000
654321軸に対応します。

【文例】

```
1 Mov P1
2 CmpG 0.0,0.0,1.0,1.0, , , ,        ' 柔らかさの設定
3 Cmp Jnt,&B11                       ' J1 軸と J2 軸を柔らかい状態にする
4 Mov P2
5 HOpen 1
6 Mov P1
7 Cmp Off                            ' 通常状態に戻す
```

【解説】

(1) 関節座標系において軸単位で、ロボットを柔らかく制御することができます。例えば、水平多関節のロボットで上下方向にピン挿入する場合、J1 軸と J2 軸を柔らかくすることにより、ある程度のピン挿入がスムーズにおこなえるようになります（上記の文例参照）。
(2) 柔らかさの度合いは、バネ定数を設定する CmpG 命令によって指定することができます。ロボットが RH-F シリーズの場合、水平軸である J1、J2 軸に 0.0 を指定することで、サーボフリー相当の機能（バネ定数＝ゼロ）になります（垂直軸は、0.0 を設定してもサーボフリー相当にはなりません。またこの機能で、動作リミットを越える位置に移動させたり、過大なずれ量とならないように注意してください）。なお、このサーボフリー相当の機能を使用している場合以降の (4)、(5) は機能しません。
(3) 柔らかい状態は、ロボットのプログラムの実行が完了した後も、Cmp Off 命令を実行する、または電源遮断時まで続きます。
(4) 柔らかい状態で押しつけ等をおこなうとき、各関節軸の動作リミットを越える位置には移動できません。
(5) 押付け等により、本来の目的位置と実際のロボットの位置とのずれ量が 200mm よりも大きくなるとロボットはそれ以上動作せず、プログラムの次の行へ実行が移ります。
(6) Cmp Jnt/Pos/Tool を同時に使うことはできません。つまり Cmp Jnt 命令が機能しているときに、Cmp Pos 命令または Cmp Tool 命令を実行するとエラーとなります。一旦 Cmp Off 命令で解除してから、これらの命令を実行してください。
(7) 本命令が機能しているときに、サーボオフからオンになると、ロボットの位置が変化することがありますのでご注意ください。
(8) コンプライアンスモードでのジョグ操作が可能です。ただし、T/B ではコンプライアンスモードの設定解除はできませんので本命令をプログラムで実行するか T/B のプログラムの編集画面でのダイレクト実行をするかにて設定してください。
(9) 軸指定を変更する場合は、一旦 Cmp Off 命令でコンプライアンスモードを解除してから、再度 Cmp Jnt 命令を実行してください。
(10) コンプライアンスモードは、ロボット本体の軸にのみ有効で、付加軸に指定しても機能しません。
(11) コンプライアンスモードが機能しているときに Fine 命令により位置決め完了条件を指定すると、動作によってはロボットが目的位置の位置決め完了パルスに到達することができず、動作命令の完了を待ち続けプログラムの実行が進まなくなる場合があります。コンプライアンスモードと Fine 命令を同時に使用しないでください。
(12) 関節座標系のコンプライアンスモードで動作しているときに誤差過大 1 エラー(H096n) が発生する場合、パラメータ CMPJCLL の設定値を大きくすると、エラーの発生が抑えられます。
(設定範囲：1 ～ 10)
このパラメータの設定値を徐々に大きな値へと変更し、誤差過大 1 エラーが発生しなくなる値を設定し

てください。
（この機能は、コントローラのソフトウェアバージョンが R4b 版 /S4b 版以上で使用できます。）

⚠注意　コンプライアンスモードは、Cmp Off 命令を実行する、または電源遮断時まで継続されます。実行プログラム番号の変更や、ジョグ操作をおこなう場合はご注意ください。

⚠注意　Cmp Jnt 命令でコンプライアンスモードとした状態でジョグ操作をおこなう場合は、関節ジョグモードで操作してください。他のジョグモードで操作すると、ジョグ操作とコンプライアンスモードで制御する座標系の方向が異なるため、移動させたい方向と異なる方向へロボットが動作する場合があります。

⚠注意　コンプライアンスモードの状態で位置を教示する際は、一旦サーボ OFF してからおこなってください。サーボ ON のまま教示操作をおこなうと、実際のロボットの位置ではなく、本来の指令位置が教示されるため、運転すると教示した時とは異なる位置へロボットが動作する場合がありますのでご注意ください。

【使用可能なロボット機種】

RH-F シリーズ

【関連システム状態変数】

M_CmpDst

【関連命令】

Cmp Off (コンプオフ)、CmpG (コンプゲイン)、Cmp Tool (コンプツール)、Cmp Pos (コンプポーズ)

【関連パラメータ】

CMPJCLL

Cmp Pos（コンプポーズ）

【機能】

直交座標系で軸方向を指定しロボットを柔らかく制御するモード（コンプライアンスモード）を開始します。

【書式】

Cmp □ Pos, ＜軸指定＞

【用語】

〈軸指定〉　柔らかく制御する軸をビットパターンで指定します。
1:有効､0:無効　&B000000
CBAZYX軸に対応します。

【文例】

```
1 Mov P1                              ' 部品挿入位置上空へ移動
2 CmpG 0.5, 0.5, 1.0, 0.5, 0.5, , ,   ' 柔らかさの設定
3 Cmp Pos, &B011011                   ' X,Y,A,B 軸を 柔らかい状態にする
4 Mvs P2                              ' 部品挿入位置へ移動
5 M_Out(10)=1                         ' 位置決め用チャック閉指示
6 Dly 1.0                             ' チャック閉完了待ち
7 HOpen 1                             ' ハンド開
8 Mvs, -100                           ' ツール座標系の Z- 方向へ 100mm 待避
9 Cmp Off                             ' 通常状態に戻す。
```

【解説】

(1) 直交座標系でロボットを柔らかく制御することが出来ます。例えば、上下方向にピン挿入する場合、X,Y,A,B 軸方向を柔らかくすることにより、ピン挿入がスムーズに行えるようになります。
(2) 柔らかさの度合いは CmpG 命令によって指定することが出来ます。
(3) 柔らかい状態は、ロボットのプログラムの実行が停止した後も、Cmp Off を実行する、または電源遮断時まで続きます。
(4) 柔らかい状態で押付け等を行う時、各関節軸の動作リミットを越える位置には移動できません。
(5) 指令位置と実際の位置とのずれ量は M_CmpDst によって読み出すことができます。ピンの挿入の成否などを、この変数を使用してチェックすることができます。
(6) 押付け等により、本来の目的位置と実際のロボットの位置とのずれ量が 200mm よりも大きくなるとロボットはそれ以上動作せず、プログラムの次の行へ実行が移ります。
(7) Cmp Jnt/Pos/Tool を同時に使うことはできません。つまり Cmp Pos 命令が機能しているときに、Cmp Jnt 命令または Cmp Tool 命令を実行するとエラーとなります。一旦 Cmp Off 命令で解除してから、これらの命令を実行してください。
(8) 本命令が機能しているときにサーボオフからオンになると、ロボットの位置が変化することがあります。また、重力方向を保持する軸は、多少落下することがあります。
(9) コンプライアンスモードでのジョグ操作が可能です。ただし、T/B ではコンプライアンスモードの設定解除はできませんので本命令をプログラムで実行するか T/B のプログラムの編集画面でのダイレクト実行をするかにて設定してください。
(10) 軸指定を変更する場合は、一旦 Cmp Off 命令でコンプライアンスモードを解除してから、再度 Cmp Pos 命令を実行してください。
(11) ロボットを特異点の近くで動作させると、アラームが発生したりロボットが大きく動作・振動したりする可能性がありますので、特異点の近くでは動作させないでください。もしこのような状況となった場合は、サーボオフの状態で一旦 Cmp Off 命令を実行（もしくは電源を再投入）してコンプライアンスモードを解除した後、ロボットを特異点から遠ざけてから、あらためてコンプライアンスモードを有効にしてください。
(12) コンプライアンスモードは、ロボット本体の軸にのみ有効で、付加軸に指定しても機能しません。
(13) コンプライアンスモードが機能しているときに Fine 命令により位置決め完了条件を指定すると、動作によってはロボットが目的位置の位置決め完了パルスに到達することができず、動作命令の完了を待ち続けプログラムの実行が進まなくなる場合があります。コンプライアンスモードと Fine 命令を同時に使用しないでください。

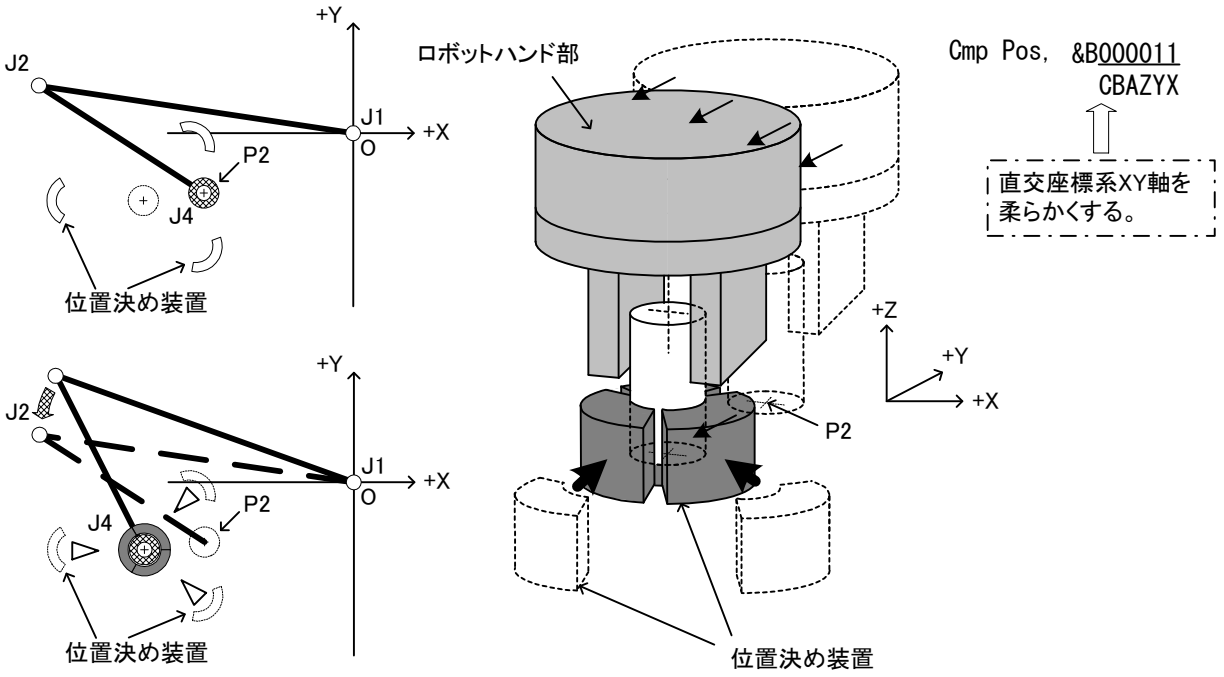


図 4-13：コンプライアンスモード使用例

⚠注意

コンプライアンスモードは、Cmp Off 命令を実行する、または電源遮断時まで継続されます。実行プログラム番号の変更や、ジョグ操作をおこなう場合はご注意ください。

⚠注意

Cmp Pos 命令でコンプライアンスモードとした状態でジョグ操作をおこなう場合は、直交ジョグモードで操作してください。他のジョグモードで操作すると、ジョグ操作とコンプライアンスモードで制御する座標系の方向が異なるため、移動させたい方向と異なる方向へロボットが動作する場合があります。

⚠注意

コンプライアンスモードの状態で位置を教示する際は、一旦サーボ OFF してからおこなってください。サーボ ON のまま教示操作をおこなうと、実際のロボットの位置ではなく、本来の指令位置が教示されるため、運転すると教示した時とは異なる位置へロボットが動作する場合がありますのでご注意ください。

【関連システム状態変数】

M_CmpDst

【関連命令】

Cmp Off（コンプオフ）、CmpG（コンプゲイン）、Cmp Tool（コンプツール）、Cmp Jnt（コンプジョイント）

Cmp Tool（コンプツール）

【機能】

ツール座標系で方向を指定しロボットを柔らかく制御するモード（コンプライアンスモード）を開始します。

【書式】

Cmp □ Tool, ＜軸指定＞

【用語】

〈軸指定〉　柔らかく制御するツール座標系の軸をビットパターンで指定します。
1:有効､0:無効　&B000000
CBAZYX軸に対応します。

【文例】

```
1 Mov P1                              ' 部品挿入位置上空へ移動
2 CmpG 0.5, 0.5, 1.0, 0.5, 0.5, , ,   ' 柔らかさの設定
3 Cmp Tool, &B011011                  ' ツール座標系の X,Y,A,B 軸を 柔らかい状態にする
4 Mvs P2                              ' 部品挿入位置へ移動
5 M_Out(10110)=1                      ' 位置決め用チャック閉指示
6 Dly 1.0                             ' チャック閉完了待ち
7 HOpen 1                             ' ハンド開
8 Mvs, -100                           ' ツール座標系の Z- 方向へ 100mm 待避
9 Cmp Off                             ' 通常状態に戻す。
```

【解説】

(1) ツール座標系でロボットを柔らかく制御することが出来ます。ツール座標系に関しては 432 ページの「5.6 標準ツール座標について」を参照願います。
(2) 例えば、ツール座標の Z 軸方向にピン挿入する場合、X,Y,A,B 軸を柔らかくすることにより、ピン挿入がスムーズに行えるようになります。
(3) 柔らかさの度合いは CmpG 命令によって指定することが出来ます。
(4) 柔らかい状態は、ロボットのプログラムの実行が停止した後も、Cmp Off を実行する、または電源遮断時まで続きます。
(5) 柔らかい状態で押付け等を行う時、各関節軸の動作リミットを越える位置には移動できません。
(6) 指令位置と実際の位置とのずれ量は M_CmpDst によって読み出すことができます。ピンの挿入の成否などを、この変数を使用してチェックすることができます。
(7) 押付け等により、本来の目的位置と実際のロボットの位置とのずれ量が 200mm よりも大きくなるとロボットはそれ以上動作せず、プログラムの次の行へ実行が移ります。
(8) Cmp Jnt/Pos/Tool を同時に使うことはできません。つまり Cmp Pos 命令が機能しているときに、Cmp Jnt 命令または Cmp Tool 命令を実行するとエラーとなります。一旦 Cmp Off 命令で解除してから、これらの命令を実行してください。
(9) 本命令が機能しているときにサーボオフからオンになると、ロボットの位置が変化することがあります。また、重力方向を保持する軸は、多少落下することがあります。
(10) コンプライアンスモードでのジョグ操作が可能です。ただし、T/B ではコンプライアンスモードの設定解除はできませんので本命令をプログラムで実行するか T/B のプログラムの編集画面でのダイレクト実行をするかにて設定してください。
(11) 軸指定を変更する場合は、一旦 Cmp Off 命令でコンプライアンスモードを解除してから、再度 Cmp Tool 命令を実行してください。
(12) 垂直 5 軸ロボット (RV-4FJL) の場合、軸指定は X 軸、Z 軸のみ可能です。
(13) ロボットを特異点の近くで動作させると、アラームが発生したりロボットが大きく動作・振動したりする可能性がありますので、特異点の近くでは動作させないでください。もしこのような状況となった場合は、サーボオフの状態で一旦 Cmp Off 命令を実行（もしくは電源を再投入）してコンプライアンスモードを解除した後、ロボットを特異点から遠ざけてから、あらためてコンプライアンスモードを有効にしてください。
(14) コンプライアンスモードは、ロボット本体の軸にのみ有効で、付加軸に指定しても機能しません。

(15) コンプライアンスモードが機能しているときに Fine 命令により位置決め完了条件を指定すると、動作によってはロボットが目的位置の位置決め完了パルスに到達することができず、動作命令の完了を待ち続けプログラムの実行が進まなくなる場合があります。コンプライアンスモードと Fine 命令を同時に使用しないでください。

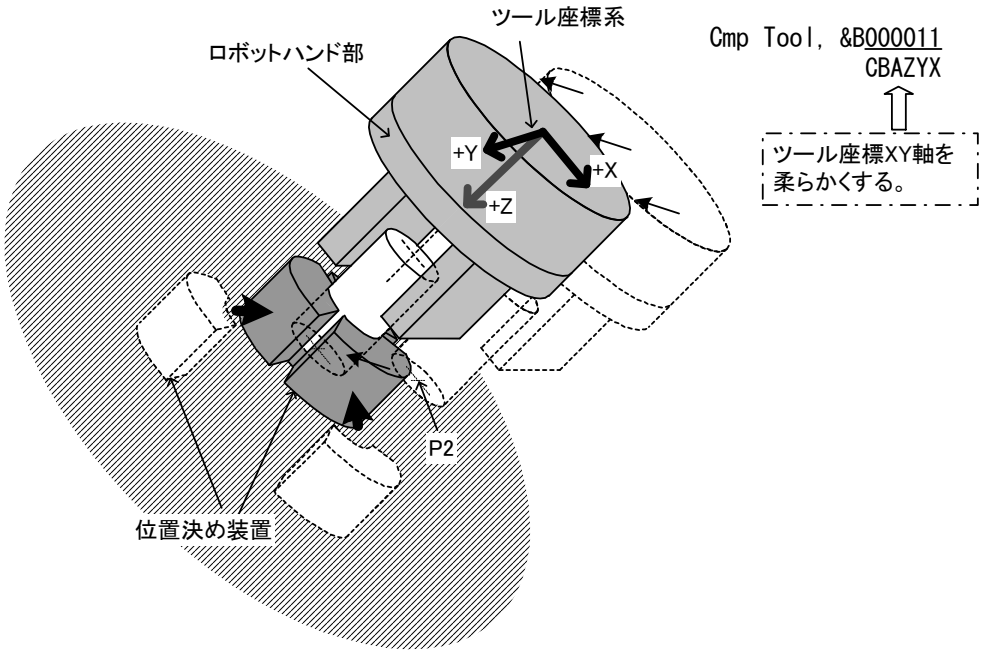


図 4-14：コンプライアンスモード使用例

⚠注意 コンプライアンスモードは、Cmp Off 命令を実行する、または電源遮断時まで継続されます。実行プログラム番号の変更や、ジョグ操作をおこなう場合はご注意ください。

⚠注意 Cmp Tool 命令でコンプライアンスモードとした状態でジョグ操作をおこなう場合は、ツールジョグモードで操作してください。他のジョグモードで操作すると、ジョグ操作とコンプライアンスモードで制御する座標系の方向が異なるため、移動させたい方向と異なる方向へロボットが動作する場合があります。

⚠注意 コンプライアンスモードの状態で位置を教示する際は、一旦サーボ OFF してからおこなってください。サーボ ON のまま教示操作をおこなうと、実際のロボットの位置ではなく、本来の指令位置が教示されるため、運転すると教示した時とは異なる位置へロボットが動作する場合がありますのでご注意ください。

【関連システム状態変数】
M_CmpDst

【関連命令】
Cmp Off（コンプオフ）、CmpG（コンプゲイン）、Cmp Pos（コンプポーズ）、Cmp Jnt（コンプジョイント）

Cmp Off（コンプオフ）

【機能】

ロボットを柔らかく制御する状態（コンプライアンスモード）を解除します。

【書式】

Cmp □ Off

【文例】

```
1 Mov P1                              ' 部品挿入位置上空へ移動
2 CmpG 0.5, 0.5, 1.0, 0.5, 0.5, , ,   ' 柔らかさの設定
3 Cmp Pos, &B011011                   ' X,Y,A,B 軸を 柔らかい状態にする
4 Mvs P2                              ' 部品挿入位置へ移動
5 M_Out(10110)=1                      ' 位置決め用チャック閉指示
6 Dly 0.5                             ' チャック閉完了待ち
7 HOpen 1                             ' ハンド開
8 Mvs, -100                           ' ツール座標系の Z- 方向へ 100mm 待避
9 Cmp Off                             ' 通常状態に戻す。
```

【解説】

(1) Cmp Tool、Cmp Pos、Cmp Jnt にて開始したコンプライアンスモードを状態を解除します。
(2) コンプライアンスモードでのジョグ操作を解除するには本命令をプログラムで実行するか T/B のプログラムの編集画面でのダイレクト実行をするかにて行ってください。

【関連命令】

CmpG（コンプゲイン）、Cmp Tool（コンプツール）、Cmp Pos（コンプポーズ）、Cmp Jnt（コンプジョイント）

CmpG（コンプゲイン）

【機能】
ロボット制御の柔らかさを指定します。

【書式】
Cmp Pos、Cmp Tool 時

```
CmpG □ [<X 軸ゲイン >], [<Y 軸ゲイン >], [<Z 軸ゲイン >], [<A 軸ゲイン >],
      [<B 軸ゲイン >], [<C 軸ゲイン >],,
```

Cmp　Jnt 時

```
CmpG □ [<J1 軸ゲイン >], [<J2 軸ゲイン >], [<J3 軸ゲイン >],
      [<J4 軸ゲイン >], [<J5 軸ゲイン >], [<J6 軸ゲイン >],,
```

【用語】
〈X～C軸ゲイン〉
〈J1～J6軸ゲイン〉　定数にて指定します。
各軸毎に柔らかさを指定することができます。
1.0が通常の状態で､最低値は機種によって違います。
各機種の最低値は【解説】(7)を参照ください。
値を省略した軸は現在の設定値を維持します。

【文例】
1 CmpG　,　,0.5, , , , ,　　'Z 軸のみを設定します。省略軸は空欄でカンマのみ記述してください。

【解説】
(1) 各軸単位で柔らかさを指定することが出来ます。
(2) Cmp Pos、Cmp Tool 命令で有効状態にしないと実際に柔らかい制御状態にはなりません。
(3) 指令位置と実際の位置とのずれに比例して、バネのように力が発生します。CmpG はそのバネ定数を指定するものです。
(4) 指令位置と実際の位置とのずれ量は M_CmpDst によって読み出すことができます。ピンの挿入の成否などを、この変数を使用してチェックすることができます。
(5) ゲインを小さくすると Cmp Pos、Cmp Tool、Cmp Jnt 命令で柔らかい制御状態にした際に、ロボットの位置が下がることがありますので、徐々に柔らかい状態にし、確認しながら設定してください。
(6) 柔らかい制御状態で本命令によりゲインの値を大きくすると、本来の目的位置と実際のロボットの位置とのずれ量を小さくするよう制御が働くため、ロボットが動作します。ゲインの値を大きくするときは、一旦 Cmp Off 命令によりコンプライアンスモードを解除してから設定するようにしてください。
(7) ゲイン値を最低値以下にしても最低値として制御します。
また､ゲイン値は小数点2桁以上が設定可能です｡各機種の最低値は以下の通りです。

機種	Cmp　Pos、Cmp　Tool 時	Cmp　Jnt 時
RH-F シリーズ	0.20, 0.20, 0.20, 0.20, 0.20, 0.20	0.00, 0.00, 0.20, 0.00, 1.00, 1.00
RV-F シリーズ	0.01, 0.01, 0.01, 0.01, 0.01, 0.01	使用不可

(8) コンプライアンスモードは、ロボット本体の軸にのみ有効で、付加軸(J7,J8 または L1,L2)に指定しても機能しません。

Cnt（コント）

【機能】

補間の連続動作を指定します。連続動作をさせることにより、動作時間の短縮が可能です。

【書式】

Cnt □ <1/0>[, <数値 1>][, <数値 2>]

【用語】

<1/0>　連続動作､加減速動作を指定します｡指定は定数でおこないます。
1:連続動作指定｡補間と補間を円滑に連続動作させます。
補間接続箇所では､速度の低下は少なく､軌跡は内回りになります。
0:加減速動作(初期値)｡補間の目的位置に到達してから､次の補間をおこないます。
補間接続箇所では､一旦速度が低下します。

<数値1>　軌跡乗り換え時の次補間開始の最大近傍距離をmm単位で指定します。
定数または変数が使用可能です。

<数値2>　軌跡乗り換え時の前補間終了の最大近傍距離をmm単位でしています。
定数または変数が使用可能です。

【文例】

軌跡乗り換え時の最大近傍距離を指定した場合

```
1 Cnt 0            ' Cnt(連続動作)を無効にする
2 Mvs P1           '加減速ありで動作する
3 Cnt 1            ' CNT(連続動作)を有効にする(この行以降の補間で連続動作)
4 Mvs P2           '次の補間との接続は連続動作
5 Cnt 1,100,200    '開始側 100mm、終了側 200mm で連続動作指定
6 Mvs P3           '補間の前後で指定距離で連続動作
7 Cnt 1,300        '開始側 300mm、終了側 300mm で連続動作指定
8 Mov P4           '開始側 300mm で連続動作
9 Cnt 0            ' Cnt(連続動作)を無効にする
10 Mov P5          '加減速ありで動作する
```

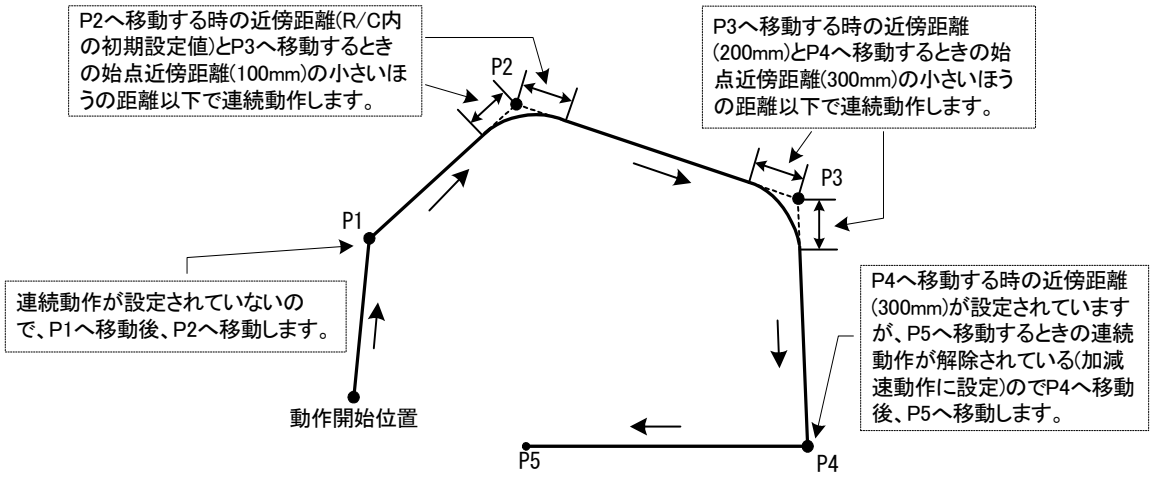


図 4-15：連続軌跡動作の例

⚠注意　ロボットは、その時の指定速度によって異なる軌跡で動作する場合があります。
特に、コーナー部分は内回り距離が変化する場合がありますので、自動運転を始める時は最初、低速で動作させ、周辺装置との干渉に注意しながら徐々に速度を上げてください。

【解説】

(1) Cnt 1 ～ Cnt 0 で囲まれた補間 (文例ステップ番号 4 ～ 8) が、連続動作の対象になります。
(2) システムの初期値は、Cnt 0(非連続動作) です。
(3) 数値 1、数値 2 を省略した場合、減速開始位置から次の補間との接続を開始します。
(4) 図 4-16 に示すように、加減速動作モードでは、目的位置手前で減速をおこない、目的位置へ移動後次の目的位置への加速を開始します。それに対し、連続動作モードは、目的位置の手前で減速はしますが、完全には停止せず、次の目的位置への移動のための加速を開始します。そのため、各目的位置を通過するのでなく、その近傍を通過します。

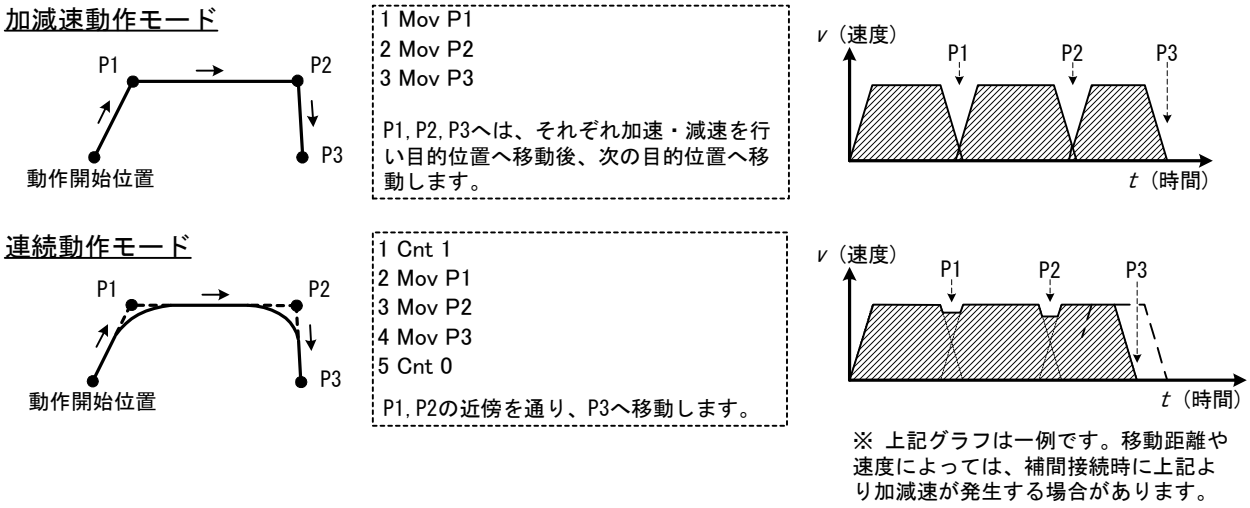


図 4-16 : 加減速動作と連続動作

(5) 近傍距離とは、次の目的位置への補間動作への乗り換え距離のことです。この近傍距離 (数値 1、数値 2) を省略した場合、加減速開始位置が次の補間への乗り換え位置となります。この場合、目的位置から離れた位置を通過しますが、動作時間が最も短くなります。目的位置のより近くを通過したい場合は、この近傍距離 (数値 1、数値 2) を設定してください。

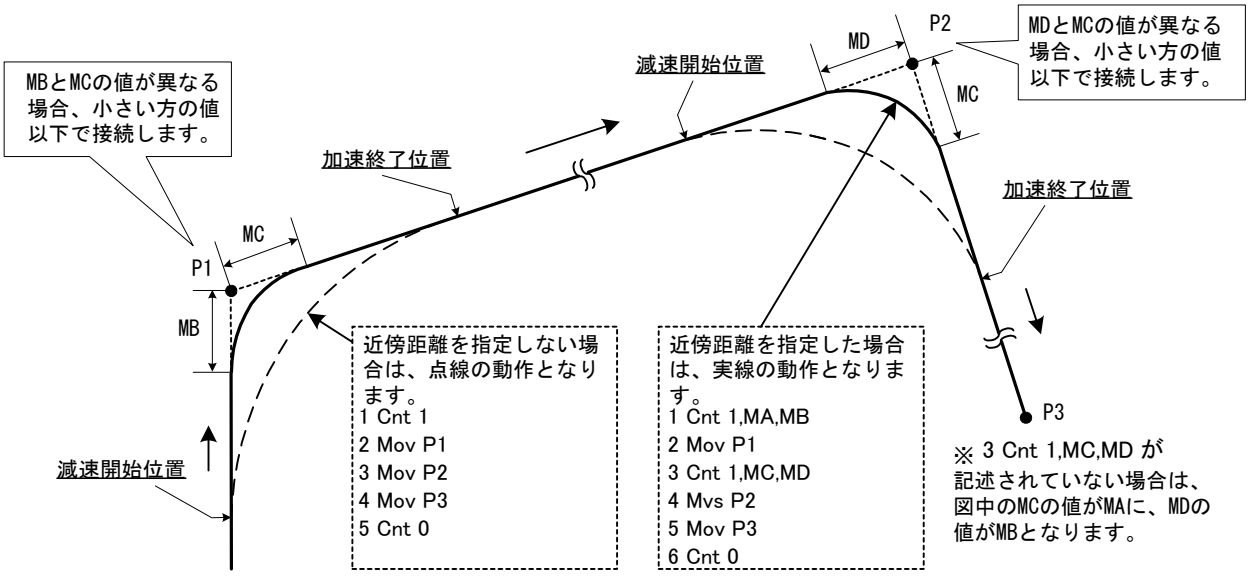


図 4-17 : 近傍距離の設定

(6) 数値 1、数値 2 の指定が異なる場合、小さい方の位置 (距離) で連続動作します。
(7) 数値 2 を省略した場合、数値 2 には数値 1 と同じ値が設定されます。
(8) 連続動作指定時には、Fine 命令による位置決め完了指定は無効になります。
(9) 近傍距離 (数値 1、数値 2) を小さくすると、Cnt 0 の状態よりも動作時間が長くなる場合があります。
(10) 連続動作指定時であっても、補間方式に特異点通過を指定した補間命令では加減速動作となります。

ColChk（コルチェック）

【機能】

自動運転における衝突検知機能の有効 / 無効を設定します。

衝突検知機能はロボットのハンドやアームが周辺機器に干渉したときにロボットを素早く停止し、ロボットのツール部分や周辺機器の破損や変形を最小限に抑えます。ただし、100% 阻止できるものではありません。

【書式】

ColChk □ On[, NOErr]/Off

【用語】

On	衝突検知を有効にします。衝突を検知したら緊急停止し、エラー1010番台を発行しサーボオフします。
Off	衝突検知を無効にします。
NOErr	衝突を検知してもエラーを発行しません。(省略時はエラーになります)

【文例 1】

衝突したらエラーにする場合

```
1 ColLvl 80,80,80,80,80,80,,        ' 衝突検知の許容レベルを指定
2 ColChk On                         ' 衝突検知を有効にする
3 Mov P1
4 Mov P2
5 Dly 0.2                           ' 動作の完了まで待つ (Fine 命令でも OK です )
6 ColChk Off                        ' 衝突検知を無効にする
7 Mov P3
```

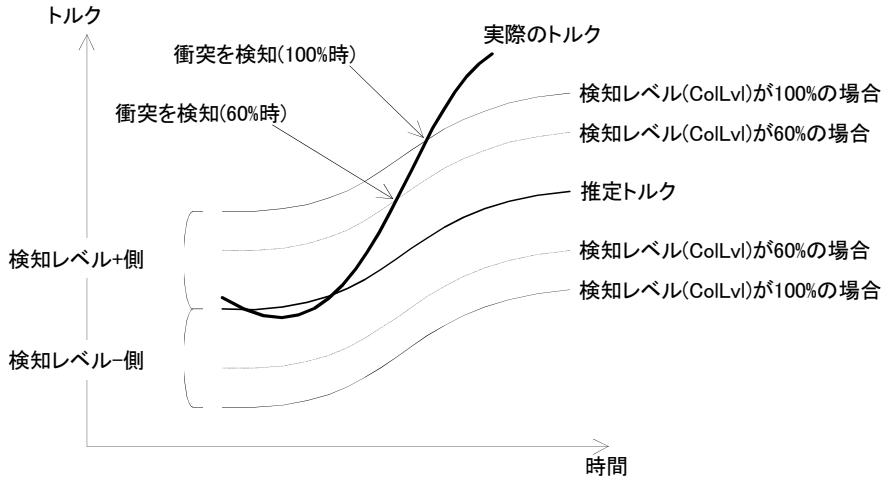
【文例 2】

衝突したら割り込み処理を使用する場合

```
1 Def Act 1,M_ColSts(1)=1 GoTo *HOME,S   ' 衝突を検知したときに実行する処理を割込で定義
2 Act 1=1
3 ColChk On,NOErr                   ' エラー非発生モードで衝突検知を有効にする
4 Mov P1
5 Mov P2                            '40 ～ 70 行実行中に衝突を検知したら割込処理へジャンプする
6 Mov P3
7 Mov P4
8 ColChk Off
9 Act 1=0
    :
100 *HOME                           ' 衝突検知時の割込処理
101 ColChk Off                      ' 衝突検知を無効にする
102 Servo On                        ' サーボ ON する
103 PESC=P_ColDir(1)*(-2)           ' 待避動作の移動量を作成
104 PDST=P_Fbc(1)+PESC              ' 待避位置を作成する
105 Mvs PDST                        ' 待避位置へ移動
106 Error 9100                      ' ユーザー定義の L レベルエラーを発生させて運転停止
```

【解説】

(1) 衝突検知とは動作命令で移動中にどの程度軸にトルクがかかるかを推定し、推定トルクと実際のトルクとの差が許容範囲を越えた場合に衝突したと判断しロボットを緊急停止する機能です。



(2) *電源 ON 直後は衝突検知無効の状態です*。パラメータ COL を有効に設定してご使用ください。
また、本命令はプログラム運転 (ステップ送り、ステップジャンプを含む) における衝突検知の有効 / 無効を指定するものです。中断中やジョグ操作中など、プログラムを実行していない時の有効 / 無効状態は、パラメータ COL の要素 3 の設定に依存します。
(3) 検知レベルは ColLvl 命令で調整可能です。検知レベルの初期値はパラメータ COLLVL の設定値になります。
(4) 本命令で衝突検知を有効とした後は、ColChk Off 命令で無効とするか、プログラムリセット、End 命令実行、電源 Off しない限り、その状態は継続されます。
(5) 本命令で衝突検知を無効にしても、ColLvl 命令で設定された衝突検知レベルは保持されます。
(6) コンティニュー機能有効時は、電源を OFF しても、次回電源 ON で以前の衝突検知設定状態が復元されます。
(7) NOErr(エラー非発生モード) を指定するとき、状態変数 M_ColSts による割り込み (割り込み条件が M_ColSts(*)=1 の割り込み。* はメカ番号) が有効になっていないと、エラー (L3950) が発生します。【文例 2】を参照ください。また、エラー非発生モード中にこの割込処理を無効にすると、エラー (L3960) が発生します。
(8) エラー非発生モードで衝突を検知すると、ロボットはサーボ OFF して停止しますがエラーとはならず運転も中断しません。ただし、エラー履歴には衝突を検知したことが記録されます。(履歴への記録は、他のエラーが併発していない場合のみ)
(9) 衝突検知機能が使用できないロボットで ColChk On と ColChk On,NOErr を実行すると、エラー (L3970) が発生します。ColChk Off の場合はエラーは発生せず無処理となります。
(10) Cmp 命令でのコンプライアンス有効中や Torq 命令でのトルク制限有効中は衝突検知を有効にできません。この時、衝突検知を有効にしようとすると、エラー (L3940) が発生します。逆に、衝突検知有効中に Cmp 命令や Torq 命令を有効にしようとすると、エラー (L3930) が発生します。
(11) 動作命令直後に ColChk Off を記述すると、動作の最後の停止付近で衝突検知できない場合があります。文例 1 のように動作命令と ColChk Off 命令の間に Dly や Fine 命令で位置決めを完了させてから ColChk Off してください。
(12) ハンド質量 (HNDDATn パラメータ) やワーク質量 (WRKDATn パラメータ) が正しく設定されていないと衝突検知が正しく働かない場合があります。パラメータを正しく設定してご使用ください。
(13) 本命令により衝突検知機能を有効にすると、プログラムによっては実行時間 (タクトタイム) が長くなる場合があります。プログラム全体に渡って衝突検知機能を有効とするのではなく、周辺機器と干渉の可能性がある動作に対してのみ、使用するようにしてください。
(14) 本機能は、マルチメカ制御と同時に使用することはできません。

【関連システム状態変数】
M_ColSts、J_ColMxl、P_ColDir
【関連命令】
ColLvl (コルレベル)
【関連パラメータ】
COL、COLLVL、COLLVLJG、HNDDATn、WRKDATn

ColLvl（コルレベル）

【機能】
自動運転における衝突検知機能の検知レベルを設定します。

【書式】

```
ColLvl  [<J1 軸 >], [<J2 軸 >], [<J3 軸 >], [<J4 軸 >],
                [<J5 軸 >], [<J6 軸 >],,
```

【用語】
〈J1～J6軸〉　検知レベルを%で指定します1～500%､省略時は前の値を保持します。
J7,J8軸は機能しません。
初期値はパラメータCOLLVLの設定値となります。

【文例】

```
1 ColLvl 80,80,80,80,80,80,,        ' 衝突検知の許容レベルを指定
2 ColChk On                         ' 衝突検知を有効にする
3 Mov P1
4 ColLvl ,50,50,,,,,                ' 衝突検知の J2,J3 軸許容レベルを変更
5 Mov P2
6 Dly 0.2                           ' P2 へ到達ご衝突検知を無効にする。
7 ColChk Off                        ' 衝突検知を無効にする
8 Mov P3
```

【解説】
(1) プログラム運転時における衝突検知機能の各軸衝突許容レベルを設定します。
(2) 本命令は自動運転（ステップ送り、ステップジャンプ操作を含む）における衝突検知機能に働きます。プログラムを運転していない時（中断中、ジョグ操作中）は、COLLVLJG パラメータの設定レベルに従います。
(3) 通常電源 ON 直後の許容レベル設定値は、パラメータ COLLVL の設定値となっています。パラメータ COLLVL の初期値は、機種毎にそれぞれ異なった値が設定されています。
(4) 値を大きくすると検知レベル（感度）が鈍り、小さくすると検知レベルが上がります。
(5) 検知レベルを上げると誤検知する確率が高くなりますので上げすぎないように調整してください。また、姿勢や動作速度によっては初期値でも誤検知する場合があります。その場合は検知レベルを下げてご使用ください。
(6) コントローラのソフトウェアバージョン R6b 版 /S6b 版以上、かつ RT ToolBox2 Ver. 3.40S 以上をご使用の場合、RT ToolBox2 のオシログラフ機能で「衝突検知レベル参考値」を選択することができます。これにより、ColLvl 命令設定値の調整作業の容易化が図れます。詳細は 472 ページの「5.21 衝突検知機能について」を参照ください。
(7) ハンド質量 (HNDDATn パラメータ) やワーク質量 (WRKDATn パラメータ) が正しく設定されていないと衝突検知が正しく働かない場合があります。パラメータを正しく設定してご使用ください。
(8) コンティニュー機能有効時は、電源を OFF しても、次回電源 ON で以前の設定値が復元されます。
(9) プログラムリセット、End 命令実行時に、許容レベルはパラメータ COLLVL の設定値にリセットされます。
(10) 衝突検知機能が使用できないロボットで本命令を実行しても、エラーにならずに無視します。
(11) J7、J8 軸に対しては、衝突検知は働きません。
(12) 同一機種でもロボットの個体差により、設定値が変わる場合があります。それぞれのロボットで動作の確認をしてください。

【関連システム状態変数】
M_ColSts、J_ColMxl、P_ColDir
【関連命令】
ColChk（コルチェック）
【関連パラメータ】
COL、COLLVL、HNDDATn、WRKDATn
※472 ページの「5.21 衝突検知機能について」に詳細を解説しています。合わせて参照願います。
また、J_ColMxl には衝突検知レベルを自動設定するサンプルプログラムを示します。

Com On/Com Off/Com Stop (コム オン / オフ / ストップ)

【機能】

Com On　　: 通信回線からの割り込みを許可します。
Com Off　　: 通信回線からの割り込みを禁止します。
Com Stop　: 通信回線からの割り込みを一時的に停止します (データは受信します)。
次回の Com On で即割り込みジャンプします。

【書式】

```
Com[(< ファイル番号 >)] □ On
```

```
Com[(< ファイル番号 >)] □ Off
```

```
Com[(< ファイル番号 >)] □ Stop
```

【用語】

〈ファイル番号〉　　通信回線に割り当てられた1～3の番号を記述します。(省略時は1)

【文例】

252 ページの「On Com GoSub (オン コム ゴーサブ)」を参照してください。

【解説】

(1) Com Off 実行後は、通信があっても割り込みは発生しません。
(2) ファイル番号については、255 ページの「Open (オープン)」を参照してください。
(3) Com Stop 実行後は、通信があっても割り込みは発生しませんが、受信データと、割り込みがあったことは記憶されていますので、Com On による割込再開時に、直ちに割込処理を実行します。

Def Act（デフ アクト）

【機能】

割り込み処理の、割り込み条件と処理を定義します。割り込みは、プログラム実行中に入力信号等を監視し、指定した条件になったときに優先的に処理を実行させたいときに使用します。

【書式】

Def □ Act □＜優先番号＞, ＜式＞□＜処理＞[, ＜タイプ＞]

【用語】

＜優先番号＞　割り込みの優先番号。1～8までの番号を定数で設定します。

＜式＞　割り込みの条件として､以下の書式で記述します。
＜数値型ﾃﾞｰﾀ＞ ＜比較演算子＞ ＜数値型ﾃﾞｰﾀ＞ または、
＜数値型ﾃﾞｰﾀ＞ ＜論理演算子＞ ＜数値型ﾃﾞｰﾀ＞
※＜数値型ﾃﾞｰﾀ＞とは以下のもの指します。
＜数値型定数＞|＜数値変数＞|＜数値配列変数＞|＜成分ﾃﾞｰﾀ＞

＜処理＞　割り込み発生時の処理を行うためのGoTo文またはGoSub文を記述します。

＜タイプ＞　省略：停止タイプ1
外部オーバーライド100%実行時を想定した停止位置で止まります。
外部オーバーライドが小さいとき停止までの時間が延びますが常に同じ位置で止まります。

S ： 停止タイプ2
外部オーバーライドに依存しないで､最短時間･最短距離で減速停止します。

L ： 実行完了停止
目的位置まで動作した(実行中の1行が完了)後に割りこみ処理をおこないます。

【文例】

```
1 Def Act 1,M_In(17)=1 GoSub *L100      ' 汎用入力信号 17 番の入力信号が ON 状態になったら、
                                        ' *L100 （ステップ番号 10）のサブルーチンを呼び出すことを定義する。
2 Def Act 2,MFG1 And MFG2 GoTo *L200    ' MFG1 と MFG2 の論理積が真になったら、
                                        ' *L200 （ステップ番号 20）へジャンプすることを定義する。
3 Def Act 3,M_Timer(1)>10500 GoSub *LBL ' 10.5 秒経過後 *LBL （ステップ番号 30）のサブルーチンにジャンプする。
  :
10 *L100:M_Timer(1)=0                   ' タイマーをゼロにセット。
11 Act 3=1                              ' Act 3 を有効にする。
12 Return 0
  :
20 *L200:Mov P_Safe
21 End
  :
30 *LBL
31 M_Timer(1)=0                         ' タイマーをゼロに戻す。
32 Act 3=0                              ' Act 3 を無効にする。
32 Return 0
```

【解説】

(1) 割込みで呼ばれた飛び先の処理の最後に Return 命令を記述します。
(2) 割り込み処理から Return 1 で次のステップへ戻る場合は、割り込み処理内で割り込み禁止にしてください。割り込み禁止にせず、割り込み条件が成立したままの場合、再度割り込み処理を実行し次のステップへ戻るため、ステップが実行されずスキップすることがあります。割り込み処理に関しては 166 ページの「Act（アクト）」を参照してください。
(3) 割り込みの優先順位は＜優先番号＞で決まり、1‥8 の順に数字が小さいほど優先順位が高くなります。
(4) 割り込みの設定は、同時に 8 個まで可能です。＜優先番号＞により区別します。
(5) ＜式＞には単純な論理演算か比較演算のみが使用可能です（演算子が 1 つ）カッコも使用できません。

(6) 同じ優先番号 Def Act 命令を記述した場合は後に定義した方が有効となります。
(7) Def Act 命令は割り込みの定義のみをおこないますので、割り込みの許可 / 禁止は Act 命令によって指定してください。
(8) 通信割り込み (Com) はどの Def Act 命令で定義した割り込みよりも優先順位が高くなります。
(9) Def Act 命令は定義したプログラム内でのみ有効です。CallP 命令 (プログラム間コール) でコールされたプログラム (ｻﾌﾞﾌﾟﾛｸﾞﾗﾑ) 中で割り込み処理を使用する場合はサブプログラム上で再定義が必要です。
(10) Def Act 命令で < 処理 > に GoTo 命令を指定した場合に割り込みが発生すると、以降のプログラム実行中は割り込み処理中のままとなり、優先順位の高い割り込みのみ受け付けます。GoTo 命令での割り込み処理中は End 命令の実行で解除されます。
(11) < 式 > には (M1 And &H001)= 1 のような論理演算との組み合わせの条件式は記述できません。
(12) 円、または円弧補間 (Mvc,Mvr,Mvr2,Mvr3) の実行中に割り込みが入り、Return 0 で元のステップに制御を戻した場合、ロボットは円、または円弧の始点に戻ってから再度円、円弧補間を実行します。
(13) アーチ補間の実行中に割り込みが入り、Return 0 で元のステップに制御を戻した場合、ロボットはその時の位置からアーチ補間をおこないます。

⚠注意　割り込みの停止タイプは、用途にあわせて正しく設定してください。
ロボットが移動命令を実行しているとき、割り込みによって最短時間・最短距離でロボットを停止させたい場合は、停止タイプに "S" を指定します。

ロボットが動作命令で移動中に、割り込み条件が成立したときの実行プログラムの停止タイプ 3 種類の概念図を図 4-18 に示します。

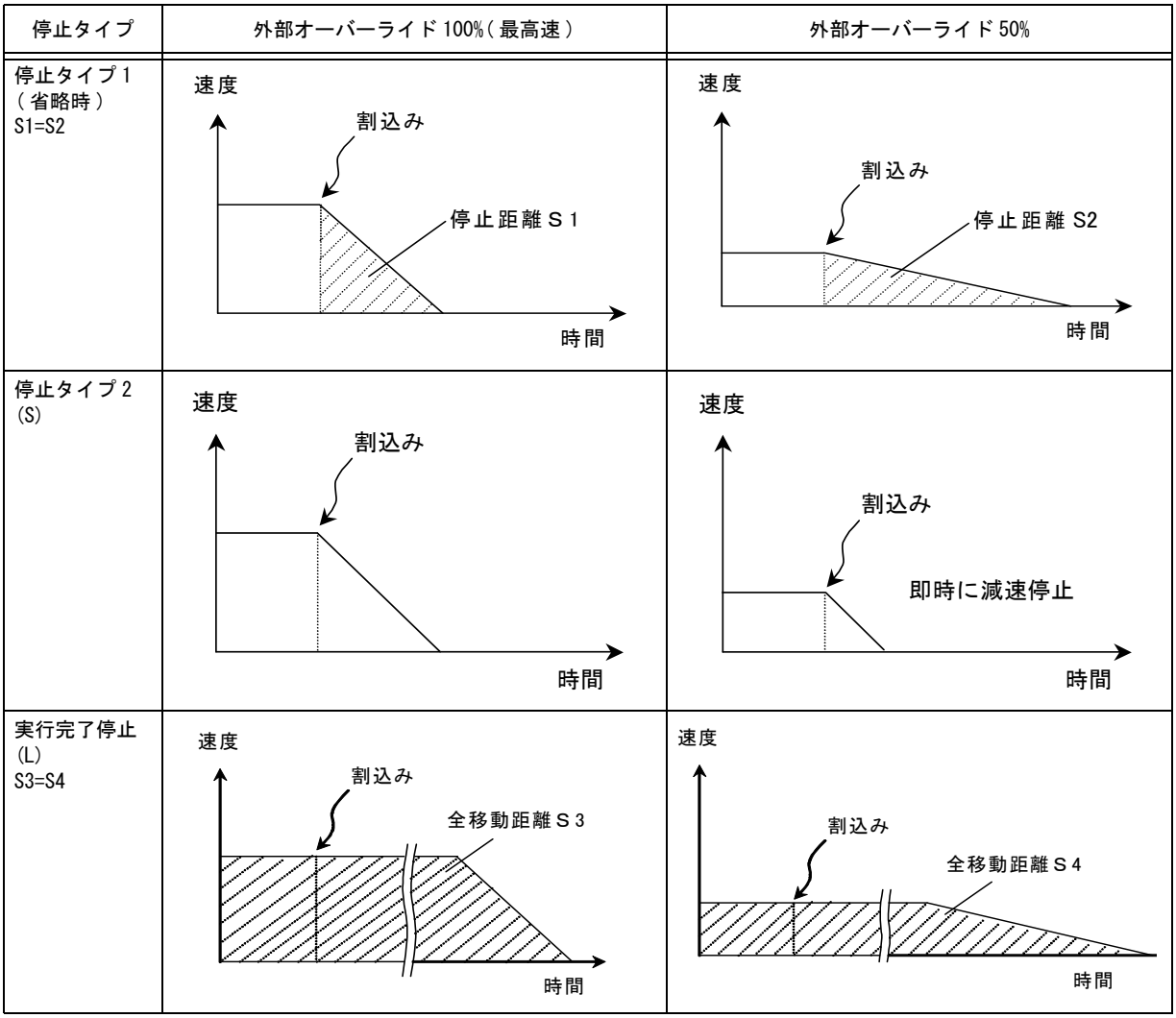


図 4-18 : 停止時の概念図

【関連命令】
Act (アクト)

Def Arch（デフ アーチ）

【機能】

Mva 命令のアーチモーション動作のアーチ形状を定義します。

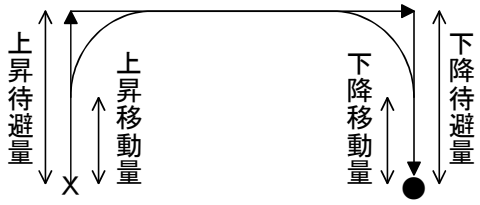
【書式】

```
Def Arch □ < アーチ番号 >, [< 上昇移動量 >], [< 下降移動量 >]
                    , [< 上昇待避量 >], [< 下降待避量 >], [< 補間形式 >]
                                        ,[< 補間タイプ 1>,< 補間タイプ 2>]
```

【用語】

〈アーチ番号〉　アーチモーション動作パターンの番号。1～4までの番号を定数または変数で設定します。

〈上昇移動量〉
〈下降移動量〉　右図照願います。定数または変数
〈上昇待避量〉　で指定できます。
〈下降待避量〉



〈補間形式〉　上昇および下降動作の補間形式
　　直線/関節 =1/0
〈補間タイプ1〉　遠回り/近回り =1/0
〈補間タイプ2〉　3軸直交/等価回転 =1/0

アーチ番号以外を省略した場合は初期値が採用されます。
初期値は下記パラメータに設定されています。値を修正することも可能です。

パラメータ名	アーチ番号	上昇移動量 (mm)	下降移動量 (mm)	上昇待避量 (mm)	下降待避量 (mm)
ARCH1S	1	0.0	0.0	30.0	30.0
ARCH2S	2	10.0	10.0	30.0	30.0
ARCH3S	3	20.0	20.0	30.0	30.0
ARCH4S	4	30.0	30.0	30.0	30.0

垂直多関節ロボット(RV-Fｼﾘｰｽﾞ)

パラメータ名	アーチ番号	補間形式	補間タイプ 1	補間タイプ 2
ARCH1T	1	1	0	0
ARCH2T	2	1	0	0
ARCH3T	3	1	0	0
ARCH4T	4	1	0	0

水平多関節ロボット(RH-Fｼﾘｰｽﾞ)

パラメータ名	アーチ番号	補間形式	補間タイプ 1	補間タイプ 2
ARCH1T	1	0	0	0
ARCH2T	2	0	0	0
ARCH3T	3	0	0	0
ARCH4T	4	0	0	0

【文例】

```
1 Def Arch 1,5,5,20,20
2 Mva P1,1          ' ステップ番号 1 で定義した形状でアーチモーション動作をします。
3 Dly 0.3
4 Mva P2,2          ' 対応する Def Arch が無い場合は、パラメータの初期値で動作します。
5 Dly 0.3
```

【解説】
(1) Def Arch 命令なしで Mva 命令を実行した場合はパラメータに設定したアーチ形状で動作します。
(2) プログラム内で上昇量などを変更する場合に使用します。

⚠注意　ロボットは、その時の指定速度によって異なる軌跡で動作する場合があります。
特に、コーナー部分は内回り距離が変化する場合がありますので、自動運転を始める時は最初、低速で動作させ、周辺装置との干渉に注意しながら徐々に速度を上げてください。

【関連命令】
Mva (ムーブアーチ)、Accel (アクセル)、Ovrd (オーバーライド)、
Mvs (ムーブエス)(補間タイプ 1、2 の参照として)

Def Char (デフ キャラ)

【機能】
文字列変数を定義します。変数名の先頭文字に "C"(または "c") 以外の文字を使用する場合に使用します。先頭に "C"(または "c") をつけた変数は Def Char で定義する必要はありません。

【書式】

Def □ Char □ < 文字列変数名 >　[, < 文字列変数名 >]...

【用語】
〈文字列変数名〉　変数名を指定します。

【文例】
```
1 Def Char MESSAGE          ' 「MESSAGE」を文字列変数として定義
2 MESSAGE = "WORKSET"       ' MESSAGE 変数に "WORKSET"を代入する。
3 CMSG = "ABC"              ' CMSG 変数に "ABC" を代入する。C で始まる変数は「Def Char」不要
```

【解説】
(1) 変数名の文字数は最大 16 文字です。使用できる文字については、137 ページの「4.4.6 プログラムで使用できる文字の種類」を参照してください。
(2) 変数名を複数指定する場合は、1 行に記述できる文字の最大値 (命令も含み 240 文字) まで設定可能です。
(3) 変数名の "C"(または "c") の次に "_" を付けユーザベースプログラムに登録するとユーザ定義外部変数 (プログラム間で共通の変数) になります。
この変数をご使用になる場合は、本命令で定義する必要があります。
詳細は 146 ページの「4.4.24 ユーザ定義外部変数」を参照ください。

Def FN（デフ ファンクション）

【機能】

任意の関数を定義します。

【書式】

Def □ FN< 識別文字 >< 名前 >[(< 仮引数 >[, < 仮引数 >]...)]=< 関数の定義式 >

【用語】

<識別文字>	識別文字は以下の4種類があります。 数値型 :Mまたはm 文字列型:Cまたはc 位置型 :Pまたはp 関節型 :Jまたはj
<名前>	任意の文字列で記述します。(最大5文字)
<仮引数>	関数が呼び出された時に、関数に引き渡たすものです。すべての変数を記述することができ、最大16変数の記述ができます。
<関数の定義式>	関数として演算を行うための式を記述します。

【文例】

```
1 Def FNMAve(ma,mb)=(ma+mb)/2        ' 二つの数値の平均を求める FNMAve を定義する。
2 MDATA1=20
3 MDATA2=30
4 MAVE=FNMAve(MDATA1,MDATA2)         ' 20 と 30 の平均値 25 が数値変数 MAVE に代入する。
5 Def FNpAdd(PA,PB)=PA+PB            ' 位置型の加算
6 P10=FNpAdd(P1,P2)
```

【解説】

(1) FN+< 名前 > が関数名となります。関数名は FN を含んで最大 16 文字（$ は含みません）までとします。
例）数値型 ･･･ FNMMAX 認識文字：M
文字列型･･･ FNCAME$ 認識文字：C(語尾に $ を記述します)

(2) Def FN で定義した関数をユーザ定義関数と呼びます。関数の定義式は、一行で記述できる範囲とします。

(3) 関数の定義式の中で、組み込み関数や既に定義済みのユーザ定義関数を記述できます。この場合、16段まで階層的にユーザ定義関数を記述できます。

(4) 関数の定義式の中で使用されている変数が、仮引数の中にない場合は、その変数がその時点で持っている値が使用されます。また、実行時に引数の型（数値または文字列）または数が宣言した型または数と異なっている場合は、エラーとなります。

(5) ユーザ定義関数は、定義したプログラムのみで有効です。CallP 命令で呼び出されたサブプログラム中では使用できません。

Def Float/Def Double/Def Inte/Def Long
(ﾃﾞﾌ ﾌﾛｰﾄ/ﾀﾞﾌﾞﾙ/ｲﾝﾃｼﾞｬｰ/ﾛﾝｸﾞ)

【機能】

数値変数を宣言します。Inte は整数、Long は長精度整数型、Float は単精度実数、Double は倍精度実数となります。

【書式】

Def □ Inte □＜数値変数名＞ [, ＜数値変数名＞]...
Def □ Long □＜数値変数名＞ [, ＜数値変数名＞]...
Def □ Float □＜数値変数名＞ [, ＜数値変数名＞]...
Def □ Double □＜数値変数名＞ [, ＜数値変数名＞]...

【用語】

〈数値変数名〉　　変数名を指定します。

【文例】

① 整数型変数の定義

```
1 Def Inte WORK1, WORK2     ' WORK1、WORK2 を数値変数名として宣言する
2 WORK1 = 100               ' WORK1 へ値 100 を代入する
3 WORK2 = 10.562            ' WORK2 は 11 となります
4 WORK2 = 10.12             ' WORK2 は 10 となります
```

② 長精度整数型変数の定義

```
1 Def Long WORK3
2 WORK3 = 12345
```

③ 単精度型実数の定義

```
1 Def Float WORK4
2 WORK3 = 123.468           ' WORK4 は、123.468000 となります。
```

④ 倍精度型実数の定義

```
1 Def Double WORK5          '
2 WORK4 = 100/3             ' WORK5 は、33.333332061767599 となります。
```

【解説】

(1) 変数名の文字数は最大 16 文字です。使用できる文字については、137 ページの「4.4.6 プログラムで使用できる文字の種類」を参照してください。
(2) 変数名を複数指定する場合は ,1 行に記述できる文字の最大値 (命令も含み 240 文字) まで設定可能です。
(3) Inte で定義した変数は整数型になります。(-32768 ～ +32767)
(4) Long で定義した変数は長精度整数型になります。(-2147483648 ～ 2147483647)
(5) Float で定義した変数は単精度型になります。(±3.40282347e+38)
(6) Double で定義した変数は倍精度型になります。(±1.7976931348623157e+308)

Def IO（デフアイオー）

【機能】

入出力変数を定義します。入出力信号のビット幅を特定したい場合に使用します。
1ビット、8ビット、16ビットの入出力信号は、標準変数で使用可能です。
(1ビット：M_In, M_Out、8ビット：M_Inb/M_In8, M_Outb/M_Out8、16ビット：M_Inw/M_In16, M_Outw/M_Out16、32ビット：M_In32, M_Out32)
本命令で宣言した変数では出力信号の参照はできませんのでご注意ください。

【書式】

Def □ IO □＜入出力変数名＞ ＝ ＜型指定＞, ＜入出力信号番号＞
[, ＜マスク情報＞]

【用語】

＜入出力変数名＞	変数名を指定します。
＜型指定＞	Bit(1bit)、Byte(8bit)、Word(16bit)のいずれかを指定
＜入出力信号番号＞	入力信号(参照時)または出力信号番号(代入時)を指定します。
＜マスク情報＞	特定の信号だけ有効にする場合に指定します。

【文例】

① PORT1という名前の入力変数を、Bit型で入出力信号6に割り当てる。

```
 1 Def IO PORT1 = Bit,6
     :
10 PORT1 = 1                ' 出力信号番号 6 が ON する。
     :
20 PORT1 = 0                ' 出力信号番号 6 が OFF する。
21 M1 = PORT1               ' 入力信号番号 6 の状態が M1 に設定される
```

② PORT2という名前の入出力変数を、Byte型で入出力信号番号5に割り当て、マスク情報を16進0Fと指定する。

```
 1 Def IO PORT2 = Byte, 5, &H0F   '
     :
10 PORT2 = &HFF                  ' 出力信号 5-8 が ON します。
                                   &H0F で上位 4bit 分をマスク（0 指定）しているため、4bit
                                   以上の出力信号を設定しても出力信号 5-8 のみ ON します。（
                                   出力信号 9-12 は変化しません）
     :
20 M2 = PORT2                    ' 入力信号 5-8 の値を変数 M2 に代入します。
```

③ PORT3という名前の入力変数を、Word型で入出力信号番号8に割り当て、マスク情報を16進0FFFと指定する。

```
 1 Def IO PORT3 = Word, 8, &H0FFF '
     :
10 PORT3 = 9                     ' 出力信号 8 と 11 が ON します。
     :
20 M3 = PORT3                    ' 入力信号 8-19 の値を変数 M3 に代入します。
```

【解説】

(1) この変数を参照するときは入力信号を見ます。
(2) この変数に代入するときは出力信号に書き込みます。
(3) この変数で出力信号を参照することはできません。出力信号を参照する場合はM_Out変数にて参照ください。
(4) 変数名の文字数は最大16文字です。使用できる文字については、137ページの「4.4.6 プログラムで使用できる文字の種類」を参照してください。
(5) この変数でパルス出力を行うことはできません。パルス出力は、M_Outを用いてください。

(6) マスク情報を指定した場合は、特定の信号だけ有効にします。
例) 上記文例②の 20 行目の例は、8 ビット幅の入出力データを 16 進 OF でマスク指定しているので、以後 PORT2 を使用すると 、
・入力信号として使用する場合 (M1=PORT2)
5 ～ 8 番を入力し、9 ～ 12 番は常に 0 と見なします。

```
        12              5 番 ( 入力信号番号 )
        0 0 0 0 1 1 1 1
          無効     有効
```

・出力信号として使用する場合 (PORT2=M1)
5 ～ 8 番には今回出力するデータを出力し、9 ～ 12 番は現在出力されている状態を保持します。

```
        12              5 番 ( 出力信号番号 )
        * * * * 1 1 1 1
          ↑     ↑
現在の出力状態保持   今回の出力データ
```

Def Jnt(デフ ジョイント)

【機能】
関節型位置変数を定義します。変数名の先頭文字に "J"(または "j") 以外の文字を使用する場合に使用します。先頭に "J"(または "j") をつけた変数は Def Jnt で定義する必要はありません。

【書式】

Def □ Jnt □ < 関節変数名 > [, < 関節変数名 >]...

【用語】
< 関節変数名 > 変数名を指定します。

【文例】

```
1 Def Jnt SAFE                          ' 「SAFE」を関節変数として定義
2 Mov J1                                ' J で始まる関節型位置変数は「Def Jnt」の定義は不要
3 SAFE = (-50,120,30,300,0,0,0,0)
4 Mov SAFE                              ' SAFE へ移動する
```

【解説】
(1) 先頭文字を "J"(または "j") 以外で関節位置変数を定義する場合に使用します。
(2) 変数名の文字数は最大 16 文字です。使用できる文字については、137 ページの「4.4.6 プログラムで使用できる文字の種類」を参照してください。変数名を複数指定する場合は、1 行に記述できる文字の最大値 (命令も含み 240 文字) まで設定可能です。
(3) 変数名の "J"(または "j") の次に "_" を付けユーザベースプログラムに登録するとユーザ定義外部変数 (プログラム間で共通の変数) になります。
この変数をご使用になる場合は、本命令で定義する必要があります。
詳細は 146 ページの「4.4.24 ユーザ定義外部変数」を参照ください。

Def Plt（デフ パレット）

【機能】

パレットを定義します。(3 点パレット /4 点パレット / 円弧パレット)

【書式】

```
Def □ Plt □＜パレット番号＞, ＜始点＞, ＜終点 A>, ＜終点 B>,
                [＜対角点＞], ＜個数 A>, ＜個数 B>, ＜パレットパターン＞
```

【用語】

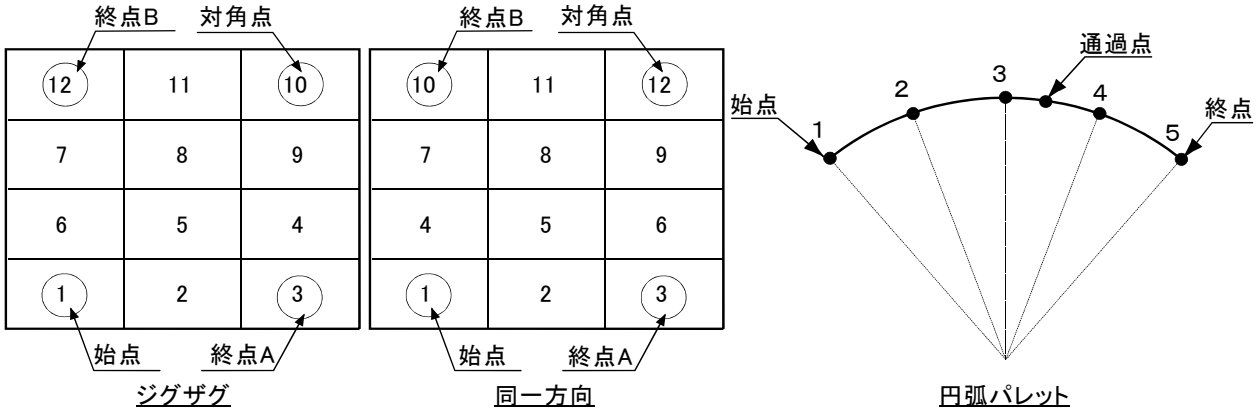
〈パレット番号〉	設定したパレットの選択番号(1～8の定数のみ)。
〈始点〉	パレットの始点。位置定数または位置変数のみ。
〈終点A〉	パレットの一方の終点。 円弧パレット時は円弧の通過点。位置定数または位置変数のみ。
〈終点B〉	パレットの他方の終点。 円弧パレット時は円弧の終点。位置定数または位置変数のみ。
〈対角点〉	パレットの始点の対角点。 円弧パレットの場合は無意味。
〈個数A〉	パレットの始点と終点A間のワークの個数。 円弧パレットの場合はパレットの始点と円弧の終点間のワークの個数。変数も可能。
〈個数B〉	パレットの始点と終点B間のワークの個数。 円弧パレットの場合は無意味(1など指定は必要)。変数も可能。
〈ﾊﾟﾚｯﾄﾊﾟﾀｰﾝ〉	分割された格子点に番号をつけるときのパレットパターンおよび姿勢の等分割/固定注)を記述します。変数も可能。 1:ジグザグ(姿勢等分割)　2:同一方向(姿勢等分割)　3:円弧パレット(姿勢等分割) 11:ジグザグ(姿勢固定)　12:同一方向(姿勢固定)　13:円弧パレット(姿勢固定)

注)姿勢等分割/姿勢固定について

Plt命令による各格子点の位置演算時に、ハンドの姿勢(向き)を等分割演算とするか、または始点の姿勢に固定するかを選択します。

姿勢等分割:始点の座標値を基準として、それぞれ終点A、終点B、及び対角点の方向に、格子点の数で等分割して演算します。

姿勢固定:　常に始点の座標値になります。



【文例】

```
1 Def Plt 1,P1,P2,P3,  ,3,4,1          '3点パレットの定義
2 Def Plt 1,P1,P2,P3,P4,3,4,1          '4点パレットの定義
```

【解説】

(1) 3 点パレットより 4 点パレットの方が位置演算時の精度が上がります。

(2) 実行したプログラム内でのみ有効です。CallP 命令で呼び出されたサブプログラム中では定義したパレット情報は使用できません。サブプログラムで使用する場合は、サブプログラム中で再定義してください。

(3) 個数 A、B は 0 でない正の数とし、0 または負の場合はエラーとなります。

(4) 個数 A× 個数 B が 32767 を超える場合は実行時エラーになります。

(5) 円弧パレットの場合、個数 B の値は意味を持ちませんが省略できませんので、0 などを設定してください。また、対角点の指定があってもそれは意味を持ちません。
(6) ハンド下向きの場合、始点、終点 A、終点 B、対角点の ABC 軸の値の符号は同一になるようにしてください。ハンドが下向きの場合は A=180(または -180)、B=0、C=180(または -180) になり、3 点の A 軸、C 軸の符号が一致しない場合は途中の位置でハンドが回転してしまう場合があります。このような場合は T/B の位置編集画面にて符号が同じになるように修正願います。+180 と -180 は同じ姿勢になり符号を修正しても問題はありません。
(7) パレットパターンに 11 ～ 13 を指定するとパレット演算で求めた位置変数の姿勢データは＜始点＞の姿勢が代入されます。1 ～ 3 を指定した場合の姿勢データは、＜始点＞-＜終点＞間を等分割したものが代入されます。
(8) J1 軸、J2 軸または J4 軸が ±180 度を越えることができるロボット機種において、J1 軸、J2 軸または J4 軸の関節角度が ±180 度をまたぐようなパレットを指定することができません。指定した場合は本命令実行時にエラー L.3750 となります。
この場合はパレットを分割してご使用願います。詳細内容を 110 ページの「4.1.2 パレット演算」に示しますので、参照願います。
(9) 本命令実行時に、指定された始点に対する終点 A、終点 B、および対角点 (省略可能) の位置関係で構成される四角形が、ロボットの動作範囲内にあるかどうかをチェックしています。
実行時にアラーム L.3110 が発生した場合は、上記四角形が動作範囲内にあることを確認してください。
(10) 一列パレットとしてご使用になる場合は、始点と終点 B を同一点 (または始点と終点 A を同一点) にして定義してください。

⚠注意

姿勢成分 (A, B, C) が ±180 度近辺の位置データをパレット定義の＜始点＞、＜終点 A, B>、＜対角点＞とする場合、位置データの同一姿勢成分に符号の異なる値が混在していると、ハンドが回転し予期せぬ動作となってしまいます。
姿勢成分が ±180 度近辺の位置データを使用する場合は、110 ページの「4.1.2 パレット演算」をご確認の上、ご使用願います。

⚠注意

パレット演算 (Plt 命令) により算出される格子点の構造フラグ (位置データの FL1) は、パレット定義の始点の値が採用されます。
したがって、パレット定義の各点で構造フラグの異なる位置データを使用すると、所望のパレット動作とは異なる動きとなってしまいます。
パレット定義の始点、終点 A, B、対角点には、構造フラグの値が全て同じとなる位置データを使用してください。また、格子点の多回転フラグ (位置データの FL2) についても、パレット定義の始点の値が採用されます。パレット定義の各点で多回転フラグの異なる位置データを使用すると、パレット動作で経由するロボットの位置や補間命令の種類 (関節補間、直線補間等) によっては、ハンドが回転し予期せぬ動作となってしまいます。そのような場合は、補間命令の引数 Type を用いて、姿勢の遠回り / 近回り動作を適切に設定し、ハンドが所望の動作となるよう調整してください。

プログラム例は、110 ページの「4.1.2 パレット演算」を参照願います。

【関連命令】
Plt (パレット)

◇◆◇パレットの姿勢について◇◆◇
パレットの格子点の姿勢を常に同じ向きにするためには、コマンドの引数の＜パレットリターン＞を 11 ～ 13 に設定してください。＜始点＞で指定した姿勢で統一されます。

Def Pos（デフ ポーズ）

【機能】

直交型位置変数を定義します。変数名の先頭文字に "P"（または "p"）以外の文字を使用する場合に使用します。先頭に "P"（または "p"）をつけた変数は Def Pos で定義する必要はありません。

【書式】

Def □ Pos □ <位置変数名> [, <位置変数名>]…

【用語】

<位置変数名> 変数名を指定します。

【文例】

```
1 Def Pos WORKSET                          ' 「WORKSET」を直交型位置変数として定義
2 Mov P1                                   ' P で始まる直交型位置変数は「Def Pos」の定義は不要
3 WORKSET=(250,460,100,0,0,-90,0,0)(0,0)
4 Mov WORKSET                              ' WORKSET へ移動する。
```

【解説】

(1) 先頭文字を "P"（または "p"）以外で直交型位置変数を定義する場合に使用します。

(2) 変数名の文字数は最大 16 文字です。使用できる文字については、137 ページの「4.4.6 プログラムで使用できる文字の種類」を参照してください。

(3) 変数名を複数指定する場合は、1 行に記述できる文字の最大値（命令も含み 240 文字）まで設定可能です。

(4) 変数名の "P"（または "p"）の次に "_" を付けユーザベースプログラムに登録するとユーザ定義外部変数（プログラム間で共通の変数）になります。
この変数をご使用になる場合は、本命令で定義する必要があります。
詳細は 146 ページの「4.4.24 ユーザ定義外部変数」を参照ください。

Dim（ディム）

【機能】
配列変数を定義します。3 次元までの配列が定義可能。

【書式】

```
Dim □ < 変数名 >(< 要素数 >[, < 要素数 >[, < 要素数 >]])
    [, < 変数名 >(< 要素数 >[, < 要素数 >[, < 要素数 >]])] …
```

【用語】

<変数名>	配列変数の名前を記述します。
<要素数>	配列変数の要素の個数を定数で記述します。

【文例】

```
1 Dim PDATA(10)              ' 10 個の要素を持った位置配列配列変数 PDATA を定義。
2 Dim MDATA#(5)              ' 5 個の要素を持った倍精度実数型配列変数 MDATA# を定義
3 Dim M1%(6)                 ' 6 個の要素を持った整数型配列変数 M1% を定義
4 Dim M2!(4)                 ' 4 個の要素を持った単精度実数型配列変数 M2! を定義
5 Dim M3&(5)                 ' 5 個の要素を持った長精度整数型配列変数 M3& を定義
6 Dim CMOJI(7)               ' 7 個の要素を持った文字列型変数 CMOJI を定義
7 Dim MD6(2,3), PD1(5,5)     ' 2×3 の要素を持った 2 次元の単精度実数型配列変数 MDATA を定義。
                             ' 5×5 の要素を持った 2 次元の位置配列変数 PD1 を定義。
```

【解説】

(1) 1 次元配列、2 次元配列、3 次元配列が定義可能です。
(2) 数値型変数の場合、変数名に変数の型を示す記号を付加することにより、整数型、長精度整数型、単精度実数型、倍精度実数型の変数を使い分けることが可能です。変数の型を省略した場合は、単精度実数型変数になります。

```
Dim MABC(10)      ' 10 個の要素を持った単精度実数型配列変数 MABC を定義
```

(3) 実際に配列変数を使用する場合の要素番号は 1 から始まります。文例のステップ番号 1 の PDATA では、要素番号は、1 ～ 10 となります。
(4) < 要素数 > は、1 ～ 999 の数値定数で記述します。数値演算式は使えません。
要素数を実数で指定した場合は、小数部を四捨五入した整数となります。
またシステムメモリの空き容量により、指定した要素数の配列が確保できない場合があります。
この場合、行登録時にエラーが発生します。
(5) 定義した要素数より大きな要素番号を使用した場合は、実行時にエラーとなります。
(6) 配列変数が定義された時点では、変数値は不定です。
(7) 配列変数を使用する場合、必ず Dim 命令による定義が必要です。
(8) Dim 命令で定義した配列は、定義したプログラムでのみ有効です。
CallP 命令で呼ばれたサブプログラム中で使用する場合は、再定義が必要となります。
(9) 配列変数は、通常の変数と同様に使用できます。
ただし、変数名および要素番号指定の文字数が 8 文字を越える変数は、ティーチングボックスの、モニタ－変数画面、および位置編集画面で使用できませんので注意が必要です。
(10) 変数名の二文字目に "_" を付けユーザベースプログラムに登録するとユーザ定義外部変数（プログラム間で共通の変数）になります。
詳細は 146 ページの「4.4.24 ユーザ定義外部変数」を参照ください。

Dly（ディレイ）

【機能】

① 単独命令の場合
指定された時間、時間待ちをします。ロボットの位置決めに使用したり、入出力信号のタイミングをとったりする場合に使用します。

② 信号出力のパルス出力用として付随的に使用する場合
パルス出力の出力時間を示します。

【書式】

①単独命令の場合

Dly □＜時間＞

②パルス出力用として付随的に使用される場合

例）M_Out(1)=1 Dly □＜時間＞

【用語】

〈時間〉 待ち時間､またはパルス出力の出力時間を定数または数値演算式で記述します｡単位:[秒]
最小値は0.01秒から設定できます｡0.00も設定できます。
最大値は単精度実数の最大値まで指定できます。

【文例】

① 時間待ち
```
1 Dly 30                ’30 秒の待ち時間をおこないます。
```
② 信号のパルス出力
```
1 M_Out(17)=1 Dly 0.5   ’0.5 秒間、汎用出力信号 17 番に信号出力（パルス出力）をおこないます。
2 M_Outb(18)=1 Dly 0.5  ’0.5 秒間、汎用出力信号 18 番に信号出力（パルス出力）をおこない、0.5 秒経過後 19-25 番に信号出力されます。
```
③ 位置決め完了を待つ
```
1 Mov P1                ’P1 へ移動
2 Dly 0.1               ’P1 への位置決めをする。
```
④ ハンド開（閉）完了を待つ
```
1 HOpen 1               ’ハンド 1 を開く
2 Dly 0.5               ’ハンド 1 の開完了を待つ
```

【解説】

(1) プログラム上での待ち時間を設定します。入出力信号のタイミングをとったり、動作命令の位置決めに使用したり、信号出力文の中に記述して（上記【文例】の②など）パルス出力時間を設定するときに使います。
(2) パルス出力は、指定時間経過を待たずに次の命令と並列実行されます。
(3) パルス出力は、同時に全プログラムで最大 50 個まで可能です。それを超えた場合には、実行時にエラーとなります。
(4) パルス出力は、指定時間経過後にビットは反転します。よって M_Outb(8 ビット）や M_Outw(16 ビット）を使用する場合はそのビット数分が反転します。
(5) パルス出力は、指定時間中に End 命令やプログラムの最終行が実行された場合には、指定時間経過を待たずにプログラムの実行は終了します。ただし、パルス出力は指定時間継続後にオフします。
(6) 他の割り込みの優先順位関係は以下の通りです。
Com ＞ Act ＞ WthIf(Wth) ＞ パルス出力（時間設定有り）
(7) パルス出力実行中に停止入力があった場合でも、パルス出力の動作は停止しません。

注 1) 下記プログラムにおいてステップ番号 2 で停止入力があった場合は、出力信号状態を保持して実行を停止します。
```
1 M_Out(17)=1
2 Dly 10
3 M_Out(17)=0
```
注 2)M_Outb(8 ビット),M_Outw(16 ビット ）を使用してパルス出力を行った場合は指定時間後にそれぞれのビット幅で反転します。
M_Outb(1)=1 Dly 1.0　この場合は 00000001 を出力し 1 秒後に 11111110 を出力します。

EMvc（イームーブシー）

【機能】

始点、通過点１、通過点２、始点の順にワーク座標系（Ex-T 座標系）に沿った三次元円補間動作を行います。

（機能概要につきましては 652 ページの「7.5 Ex-T 制御について」を参照ください。）

【書式】

EMvc □＜ワーク座標番号＞,＜始点＞,＜通過点 1＞,＜通過点 2＞[□＜付随条件＞]

【用語】

＜ワーク座標番号＞	１～８のワーク座標番号を指定します。
＜始点＞	円の始点および終点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
＜通過点1＞	円弧の通過点1。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
＜通過点2＞	円弧の通過点２。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
＜付随条件＞	Wth節またはWthIf節を記述します。

【文例】

① P1、P2、P3、P1 の順にワーク座標１に沿って円補間移動

```
1 EMvc 1, P1, P2, P3
```

② P1、J2、P3、P1 の順にワーク座標２に沿って円補間移動（関節変数を使用した例）

```
2 EMvc 2, P1, J2, P3
```

③ P1、P2、P3、P1 の順にワーク座標１に沿って円補間移動と同時に出力信号 17 をオン

```
3 EMvs 2,P1 Wth M_Out(17)=1
```

④ P3、(Plt 1,5)、P4、P3 の順にワーク座標４に沿って円補間移動し、その移動中に入力信号 20 がオンしたら出力信号 21 をオン

```
4 EMvc 4, P3, (Plt 1, 5), P4 WthIf M_In(20)=1, M_Out(21)=1
```

【解説】

(1) Ex-T 制御円補間は、与えられた 3 点から指定されたワーク座標に沿った円を求め、円周を移動します。（360 度）

(2) Ex-T 制御円補間中の姿勢は、始点の姿勢を維持します。通過点 1，2 の姿勢は考慮されません。

(3) 現在位置と始点が一致しない場合は、自動的に始点まで Ex-T 制御直線補間で移動しその後に Ex-T 制御円補間をおこないます。

(4) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで移動した後、残りの円弧補間を再開します。中断位置へ戻る補間方法（関節補間 / 直線補間）は RETPATH パラメータで変更できます（442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください）。

(5) ALWAYS、ERROR 属性のプログラムでは実行することはできません。（エラー L3287 が発生します。）

(6) ワーク座標系が設定されていない場合、その初期値は全成分 0(0, 0, 0, 0, 0, 0) となっています。この状態で Ex-T 制御直線補間を実行すると、ワールド座標系原点に沿って動作することになります。

(7) 付加軸が付加されている場合、付加軸も動作します。ただし、付加軸は Ex-T 制御に含まれません。そのため、走行軸などロボット本体を移動してしまうような場合、ワーク座標もロボットとともに移動することとなり、本来の作業位置から逸脱することが想定されます。よって本命令での付加軸動作は推奨しません。また、本命令を使用するときは、付加軸同期制御は使用しないでください。

【関連命令】

EMvr（イームーブアール）、EMvr2（イームーブアール 2）、EMvr3（イームーブアール 3）、EMvs（イームーブエス）

【関連システム状態変数】

P_WkCord（ワーク座標データ）

【関連パラメータ】

WK1CORD ～ WK8CORD

EMvr（イームーブアール）

【機能】

始点から通過点を経由して終点までワーク座標系（Ex-T 座標系）に沿った三次元円弧補間動作を行います。

（機能概要につきましては 652 ページの「7.5 Ex-T 制御について」を参照ください。）

【書式】

EMvr □＜ワーク座標番号＞,＜始点＞,＜通過点＞,＜終点＞[□ Type □＜定数 1>,＜定数 2>]

[□＜付随条件＞]

【用語】

〈ワーク座標番号〉	１～８のワーク座標番号を指定します。
〈始点〉	円弧の始点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
〈通過点〉	円弧の通過点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
〈終点〉	円弧の終点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
〈定数1〉	姿勢回転の指定。近回り動作＝0。省略時は0。
〈定数2〉	姿勢補間方法の指定。等価回転/三軸直交= 0/1。省略時は0。
〈付随条件〉	Wth節またはWthIf節を記述します。

【文例】

① P1 から P2 を経由して P3 にワーク座標 1 に沿って円弧補間移動

```
1 EMvr 1, P1, P2, P3
```

② P1 から J2 を経由して P3 にワーク座標 2 に沿って円弧補間移動（関節変数を使用した例）

```
2 EMvr 2, P1, J2, P3
```

③ P1 から P2 を経由して P3 にワーク座標 1 に沿って円弧補間移動と同時に出力信号 17 をオン

```
3 EMvr 2, P1, P2, P3 Wth M_Out(17)=1
```

④ P3 から (Plt 1,5) を経由して P4 にワーク座標 4 に沿って円弧補間移動し、その移動中に入力信号 20 がオンしたら出力信号 21 をオン

```
4 EMvr 4, P3, (Plt 1, 5), P4 WthIf M_In(20)=1, M_Out(21)=1
```

【解説】

(1) Ex-T 制御円弧補間は、与えられた 3 点から指定されたワーク座標に沿った円弧を求め、その円弧上を移動します。
(2) 姿勢は、始点から終点への補間となり、通過点の姿勢は影響しません。
(3) 現在位置と始点が一致しない場合は、自動的に始点まで Ex-T 制御直線補間で移動しその後に円弧補間をおこないます。
(4) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで移動した後、残りの円弧補間を再開します。中断位置へ戻る補間方法（関節補間 / 直線補間）は RETPATH パラメータで変更できます（442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください）。
(5) 等価回転（定数 2=0）の場合、始点と終点の構造フラグが異なると、実行時にエラーとなります。
(6) 指定した 3 点のうち同一位置がある場合や、3 点が一直線上にある場合は始点から終点に向かって Ex-T 制御直線補間動作をします。エラーとはなりません。
(7) 定数 2 に三軸直交を指定した場合、定数 1 は無効でティーチング通りの姿勢に移動します。
(8) ALWAYS、ERROR 属性のプログラムでは実行することはできません。（エラー L3287 が発生します。）
(9) ワーク座標系が設定されていない場合、その初期値は全成分 0 (0, 0, 0, 0, 0, 0) となっています。この状態で Ex-T 制御直線補間を実行すると、ワールド座標系原点に沿って動作することになります。
(10) 付加軸が付加されている場合、付加軸も動作します。ただし、付加軸は Ex-T 制御に含まれません。そのため、走行軸などロボット本体を移動してしまうような場合、ワーク座標もロボットとともに移動することとなり、本来の作業位置から逸脱することが想定されます。よって本命令での付加軸動作は推奨しません。また、本命令を使用するときは、付加軸同期制御は使用しないでください。
(11) 定数 2 に三軸直交を指定した場合、Mvr 命令にて三軸直交を指定した場合と同様の動作となります。

【関連命令】
EMvc（イームーブシー）、EMvr2（イームーブアール 2）、EMvr3（イームーブアール 3）、EMvs（イームーブエス）

【関連システム状態変数】
P_WkCord（ワーク座標データ）

【関連パラメータ】
WK1CORD ～ WK8CORD

EMvr2（イームーブアール 2）

【機能】

始点、終点、参考点から構成される円弧上を、始点から終点までワーク座標系（Ex-T 座標系）に沿った三次元円弧補間動作を行ないます。移動方向は参考点を通過しない方向となります。

（機能概要につきましては 652 ページの「7.5 Ex-T 制御について」を参照ください。）

【書式】

EMvr2 □＜ワーク座標番号＞,＜始点＞,＜終点＞,＜参考点＞[□ Type □＜定数 1>,＜定数 2>]

[□＜付随条件＞]

【用語】

〈ワーク座標番号〉	１～８のワーク座標番号を指定します。
〈始点〉	円弧の始点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
〈終点〉	円弧の終点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
〈参考点〉	円弧の参考点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
〈定数1〉	姿勢回転の指定。近回り動作＝0。省略時は0。
〈定数2〉	姿勢補間方法の指定。等価回転/三軸直交= 0/1。省略時は0。
〈付随条件〉	Wth節またはWthIf節を記述します。

【文例】

```
1 EMvr2 1, P1, P2, P3
2 EMvr2 2, P1, J2, P3
3 EMvr2 2, P1, P2, P3 Wth M_Out(17)=1
4 EMvr2 4, P3, (Plt 1, 5), P4 WthIf M_In(20)=1, M_Out(21)=1
```

【解説】

(1) 円弧補間動作は、与えられた 3 点から円弧を求め、その円弧上を移動します。
(2) 姿勢は、始点から終点への補間となり、参考点の姿勢は影響しません。
(3) 現在位置と始点が一致しない場合は、自動的に始点まで直線補間（三軸直交補間）します。
(4) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで関節補間で移動した後、残りの円弧補間を再開します。中断位置へ戻る補間方法（関節補間 / 直線補間）は RETPATH パラメータで変更できます（442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください）。
(5) 移動方向は参考点を通らない方向です。
(6) 等価回転（定数 2=0）の場合、始点と終点の構造フラグが異なる場合は、実行時にエラーとなります。
(7) 指定した 3 点のうち同一位置がある場合や、3 点が一直線上にある場合は始点から終点に向かって直線補間動作をします。エラーとはなりません。
(8) 定数 2 に三軸直交を指定した場合、定数 1 は無効でティーチング通りの姿勢に移動します。
(9) ALWAYS、ERROR 属性のプログラムでは実行することはできません。（エラー L3287 が発生します。）
(10) ワーク座標系が設定されていない場合、その初期値は全成分 0(0, 0, 0, 0, 0, 0) となっています。この状態で Ex-T 制御直線補間を実行すると、ワールド座標系原点に沿って動作することになります。
(11) 付加軸が付加されている場合、付加軸も動作します。ただし、付加軸は Ex-T 制御に含まれません。そのため、走行軸などロボット本体を移動してしまうような場合、ワーク座標もロボットとともに移動することとなり、本来の作業位置から逸脱することが想定されます。よって本命令での付加軸動作は推奨しません。また、本命令を使用するときは、付加軸同期制御は使用しないでください。
(12) 定数 2 に三軸直交を指定した場合、Mvr 命令にて三軸直交を指定した場合と同様の動作となります。

【関連命令】

EMvc（イームーブシー）、EMvr（イームーブアール）、EMvr3（イームーブアール 3）、EMvs（イームーブエス）

【関連システム状態変数】

P_WkCord（ワーク座標データ）

【関連パラメータ】

WK1CORD ～ WK8CORD

EMvr3（イームーブアール 3）

【機能】

始点、終点、中心点から構成される円弧上を、始点から終点までワーク座標に沿った三次元円弧補間動作を行ないます。

（機能概要につきましては 652 ページの「7.5 Ex-T 制御について」を参照ください。）

【書式】

EMvr3 □＜ワーク座標番号＞,＜始点＞,＜終点＞,＜中心点＞[□ Type □＜定数 1＞,＜定数 2＞]

[□＜付随条件＞]

【用語】

＜ワーク座標番号＞	１～８のワーク座標番号を指定します。
＜始点＞	円弧の始点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
＜終点＞	円弧の終点。 位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
＜中心点＞	円弧の中心点。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。
＜定数1＞	姿勢回転の指定。近回り動作＝0。省略時は0。
＜定数2＞	姿勢補間方法の指定。等価回転/三軸直交= 0/1。省略時は0。
＜付随条件＞	Wth節またはWthIf節を記述します。

【文例】

```
1 EMvr3 1, P1, P2, P3
2 EMvr3 2, P1, J2, P3
3 EMvr3 2, P1, P2, P3 Wth M_Out(17)=1
4 EMvr3 4, P3, (Plt 1, 5), P4 WthIf M_In(20)=1, M_Out(21)=1
```

【解説】

(1) 円弧補間動作は、与えられた 3 点から円弧を求め、その円弧上を移動します。
(2) 姿勢は、始点から終点への補間となり、中心点の姿勢は影響しません。
(3) 現在位置と始点が一致しない場合は、自動的に始点まで直線補間（三軸直交補間）します。
(4) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで関節補間で移動した後、残りの円弧補間を再開します。中断位置へ戻る補間方法（関節補間 / 直線補間）は RETPATH パラメータで変更できます（442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください）。
(5) 等価回転（定数 2=0）の場合、始点と終点の構造フラグが異なる場合は、実行時にエラーとなります。
(6) 定数 2 に三軸直交を指定した場合、定数 1 は無効でティーチング通りの姿勢に移動します。
(7) 始点から終点までの中心角は、0＜中心角＜180 度です。
(8) 中心点から始点までの距離と、中心点から終点までの距離の差は、0.01mm 以内になるように位置を指定してください。
(9) 3 点が同一直線上にある場合や、始点と中心点、または終点と中心点が同一の場合はエラーとなります。
(10) 始点と終点が同一の場合や 3 点が同一の場合はエラーとならず、次の命令を実行します。
ただし、この時姿勢が異なる場合は姿勢のみの補間をおこないます。
(11) ALWAYS、ERROR 属性のプログラムでは実行することはできません。（エラー L3287 が発生します。）
(12) ワーク座標系が設定されていない場合、その初期値は全成分 0(0, 0, 0, 0, 0, 0) となっています。この状態で Ex-T 制御直線補間を実行すると、ワールド座標系原点に沿って動作することになります。
(13) 付加軸が付加されている場合、付加軸も動作します。ただし、付加軸は Ex-T 制御に含まれません。そのため、走行軸などロボット本体を移動してしまうような場合、ワーク座標もロボットとともに移動することとなり、本来の作業位置から逸脱することが想定されます。よって本命令での付加軸動作は推奨しません。また、本命令を使用するときは、付加軸同期制御は使用しないでください。
(14) 定数 2 に三軸直交を指定した場合、Mvr 命令にて三軸直交を指定した場合と同様の動作となります。

【関連命令】

EMvc（イームーブシー）、EMvr（イームーブアール）、EMvr2（イームーブアール 2）、EMvs（イームーブエス）

【関連システム状態変数】
P_WkCord（ワーク座標データ）

【関連パラメータ】
WK1CORD ～ WK8CORD

EMvs（イームーブエス）

【機能】

現在位置から目的位置までワーク座標系（Ex-T 座標系）に沿った直線補間動作を行います。
（機能概要につきましては 652 ページの「7.5 Ex-T 制御について」を参照ください。）

【書式】

EMvs □＜ワーク座標番号＞,＜移動目的位置＞[□ Type □＜定数 1>,＜定数 2>]
[□＜付随条件＞]

【用語】

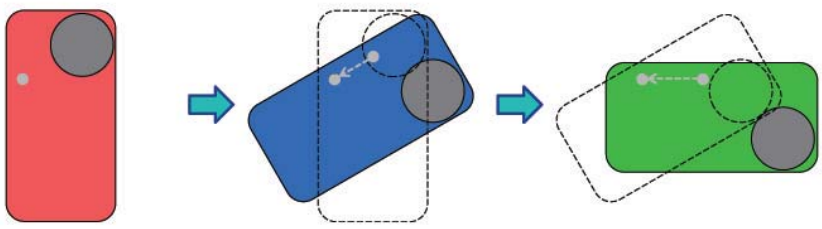
〈ワーク座標番号〉　１～８のワーク座標番号を指定します。
〈移動目的位置〉　補間動作の最終位置を指定します。位置型の変数と定数、論理/算術式と関数、または関節変数で記述します。(ロボット制御点の位置データを入力します。)
〈定数1〉　姿勢回転の指定。近回り動作= 0。省略時は0。
〈定数2〉　姿勢補間方法の指定。等価回転/三軸直交 = 0/1。省略時は0。
〈付随条件〉　Wth節またはWthIf節を記述します。

【文例】

① 目的位置 P1 へワーク座標 1 に沿って直線補間移動
```
1 EMvs 1,P1
```
② 目的位置 (Plt 1,5) へワーク座標 3 に沿って直線補間移動
```
1 EMvs 3,(Plt 1,5)
```
③ 目的位置 P1 へワーク座標 2 に沿って直線補間移動と同時に出力信号 17 をオン
```
1 EMvs 2,P1 Wth M_Out(17)=1
```
④ 目的位置 P4 へワーク座標 1 に沿って直線補間移動し、その移動中に入力信号 18 がオンしたら出力信号 20 をオン
```
1 EMvs 1,P4 WthIf M_In(18)=1, M_Out(20)=1
```

【解説】

(1) EMvs 命令（Ex-T 制御直線補間）は、指定されたワーク座標系に沿ってロボットの現在位置から移動目的位置まで移動する動作形態です。
EMvs 命令での動作例を図に示します。Ex-T 制御点（Ex-T 座標系原点）に沿って、姿勢を変えながら直線補間を行う例です。



(2) 姿勢は、ワーク座標から見た始点から終点への補間となります。
(3) ワーク座標番号 1 ～ 8 は WK1CORD ～ WK8CORD に対応します。
(4) MELFA-BASIC V のみ使用可能です。

(5) Ex-T 制御直線補間の動作速度は、Spd 命令によって指定することができます。また、Ovrd 命令および O/P オーバーライド設定値によっても変更することができます。
(6) Ex-T 制御直線補間の加減速は、Oadl On 時（初期値）では最適加減速となります。Accel 命令は有効です。
(7) MvTune 命令で動作モードを切り替えるができます。
(8) Cnt 命令による軌跡接続、Fine 命令による位置決め精度指定も可能です。
(9) Ex-T 制御直線補間の動作中の中断 / 再開は可能です。中断再開による中断点への移動は Ex-T 制御直線補間ではなく通常の補間動作となります。この中断点への移動の補間形式は、RETPATH パラメータの設定によります。初期設定は「関節補間による中断位置への移動」となっています。
(10) ALWAYS、ERROR 属性のプログラムでは実行することはできません。（エラー L3287 が発生します。）
(11) ワーク座標系が設定されていない場合、その初期値は全成分 0(0, 0, 0, 0, 0, 0) となっています。この状態で Ex-T 制御直線補間を実行すると、ワールド座標系原点に沿って動作することになります。
(12) 付加軸が付加されている場合、付加軸も動作します。ただし、付加軸は Ex-T 制御に含まれません。そのため、走行軸などロボット本体を移動してしまうような場合、ワーク座標もロボットとともに移動することとなり、本来の作業位置から逸脱することが想定されます。よって本命令での付加軸動作は推奨しません。また、本命令を使用するときは、付加軸同期制御は使用しないでください。
(13) 定数 2 に三軸直交を指定した場合、Mvs 命令にて三軸直交を指定した場合と同様の動作となります。

【関連命令】
EMvc（イームーブシー）、EMvr（イームーブアール）、EMvr2（イームーブアール 2）、EMvr3（イームーブアール 3）

【関連システム状態変数】
P_WkCord（ワーク座標データ）

【関連パラメータ】
WK1CORD ～ WK8CORD

EMvSpl（イームーブスプライン）

【機能】

指定したスプラインファイルに登録されている情報に基づき、Ex-T 座標系原点に沿ったスプライン補間を実行します。（スプライン補間の詳細については 587 ページの「7.4 スプライン補間について」を参照してください。）

注）この命令はソフトウェアバージョン Ver.R6b 以降（F-Q シリーズ）/Ver.S6b 以降（F-D シリーズ）で使用できます。

【書式】

EMvSpl □ <Ex-T 座標番号 >,< スプライン番号 >,< 速度 >,< 加減速距離 >
[,< フレーム変換 > [,< 姿勢補間タイプ > [,< フィルタ長 >]]]

【用語】

〈Ex-T座標番号〉 制御点とするEx-T座標の番号を､定数または数値変数で指定します。
設定範囲:
0:スプラインファイルに登録されているEx-T座標系原点データを使用します。
1～8:番号に対応したワーク座標系原点データを使用します。

〈スプライン番号〉 動作させたい経路の情報を保持しているスプラインファイルの番号を､定数または数値変数で指定します。
設定範囲:1～99

〈速度〉 スプライン補間の動作速度を､定数または数値変数で指定します。
設定範囲:0.0より大きな値(mm/s)

〈加減速距離〉 スプライン補間の加減速距離を､定数または数値変数で指定します。
加速距離は､動作開始から指定した速度となるまでの加速に要する距離です。
減速距離は､指定した速度から終了位置に停止するまでの減速に要する距離です。
スプライン補間では､加速距離と減速距離は共通の設定となります。
設定範囲:0.0より大きな値(mm/s)

〈フレーム変換〉 フレーム変換の実行内容を､定数または数値変数で指定します。
設定範囲:
0:フレーム変換を実行しません。
1:スプラインファイルに設定された座標系を使ってフレーム変換を実行します。
2:SetCalFrm命令で設定された座標系を使ってフレーム変換を実行します。
省略時:フレーム変換を実行しません。

〈姿勢補間タイプ〉 姿勢の補間タイプを､定数または数値変数で指定します。
設定範囲:
0(等価回転動作)/1(三軸直交動作)
省略時:等価回転動作

〈フィルタ長〉 加減速動作にかかるフィルタ長を､定数または数値変数で指定します。
設定範囲:0.0～1000.0(ms)
省略時:100.0ms

【文例】

```
1  EMvSpl 0,2,20,10     ' スプラインファイル 2 に設定された経路点データで生成されるスプライン曲線を、スプラインファイルに登録された Ex-T 座標系原点に沿わせる Ex-T スプライン補間を実行
2  EMvSpl 7,2,30,10     ' スプラインファイル 2 に設定された経路点データで生成されるスプライン曲線を、ワーク座標 7 に沿わせる Ex-T スプライン補間を実行
```

【解説】

(1) <Ex-T 座標番号 > に 1 ～ 8 を指定した場合は、Wk1CORD ～ WK8CORD に設定された座標系データに沿ったスプライン補間を実行します。0 を指定した場合は、スプラインファイルに登録されている Ex-T 座標系原点データに沿ったスプライン補間を実行します。

(2) スプラインファイルに Ex-T 座標設定がされていない場合は、エラー L2610(Ex-T 座標設定不正) が発生します。
(3) < スプライン番号 > に対応したスプラインファイルに登録されている経路点データなどの情報に基づき、Ex-T スプライン補間を実行します。
(4) 把持するワークから見た Ex-T 座標系原点が、滑らかな曲線となるように動作します。
(5) EMvSpl 命令の実行開始時のロボットの現在位置と Ex-T スプライン補間の開始位置が一致しない場合は、開始位置まで Ex-T 直線補間 (EMvs 命令) で動作してから Ex-T スプライン補間を実行します。
(6) < スプライン番号 > に対応したスプラインファイルがコントローラに保存されていない場合は、L2610(スプラインファイルを開くことができません) が発生します。
(7) Ex-T スプライン補間の指令速度 (ワークから見た Ex-T 座標系の速度) は、次式のとおり指定されます。
指令速度 = (EMvSpl 命令の < 速度 >) × (Ovrd 命令) × (操作パネルのオーバーライド)
Spd 命令と JOvrd 命令の設定は使用されません。また、< 速度 > に状態変数 M_NSpd を設定しても最適速度制御モードは機能しません。
(8) Ex-T スプライン補間の加減速は、< 加減速距離 > で指定します。Accel 命令の設定は使用されません。また、Oadl 命令で最適加減速制御を有効としていても、Ex-T スプライン補間には適用されません。
(9) 加速動作は、Ex-T スプライン補間の開始位置から < 加減速距離 > を移動すると < 速度 > に到達するように速度を生成します。減速動作は、< 速度 > から < 加減速距離 > を移動して終了位置で停止するように速度を生成します。
Ovrd 命令・操作パネルのオーバーライドが 100% より小さい値が設定されている場合は、< 加減速距離 > よりも短い移動量で指令速度に到達します。
(フィルタ等の影響により、実際には < 加減速距離 > で指定した値よりも長めの加減速距離となります。)
(10) < フレーム変換 > に「1(スプラインファイルに設定した座標系を使用して実行)」・「2(SetCalFrm 命令で設定した座標系を使用して実行)」を指定した場合は、指定された方法に基づいて逐次経路点データを変換し、その経路点を通るように Ex-T スプライン補間を実行します。
(11) < フレーム変換 > に「1(スプラインファイルに設定した座標系を使用して実行)」を指定しているにも関わらず、スプラインファイルに座標系が設定されていない場合は、エラー L2042(フレーム変換座標が設定されていません) が発生します。
(12) < フレーム変換 > に「1(スプラインファイルに設定した座標系を使用して実行)」を指定している場合に座標系が算出できない場合は、エラー L2041(フレーム変換座標が算出できません) が発生します。
(13) < フレーム変換 > に「2(SetCalFrm 命令で設定した座標系を使用して実行)」を指定すると、最後に実行した SetCalFrm 命令により設定された座標系を使ってフレーム変換を実行します。一度も SetCalFrm 命令が実行されておらず座標系が設定されていない場合は、エラー L2042(フレーム変換座標が設定されていません) が発生します。
(14) < 姿勢補間タイプ > の指定により、「0(等価回転動作)」と「1(三軸直交動作)」を切り替えることができます。
等価回転動作は、経路点間の姿勢変化が最少の角度となるように動作します。
三軸直交動作は、姿勢データを A,B,C 軸座標値ではなく J4,J5,J6 軸角度で生成します。そのため、経路点の姿勢データ (A,B,C 軸座標値) が同一であっても、動作中はロボットの姿勢が変化します。
(15) 構造フラグの乗り越え (特異点通過) には対応していません。経路点間で構造フラグが異なる場合は、エラー L2611(経路点の構造フラグが異なります) が発生します。
(16) < フィルタ長 > により加減速動作にかかるフィルタ特性を変更することができます。加速・減速時の動作をより滑らかにしたい場合は、フィルタ長を大きくします。ただし、大きくしすぎると動作が緩慢となり、Ex-T スプライン補間が終了するまでの時間が長くなります。
(17) 経路点データに信号出力が設定されている場合は、対象の経路点を通過する際、指定された条件で汎用出力信号を出力します。
(18) ロボットの位置決め完了条件 (Fine 命令、Fine J 命令、Fine P 命令) が指定されている場合、Ex-T スプライン補間の終了位置で位置決め完了を確認します。
(19) Cnt 命令により連続動作が指定されていても、Ex-T スプライン補間の開始位置・終了位置では連続動作はおこないません。
(20) Def Act 命令 (割り込み条件と処理の定義) の < タイプ > に「停止タイプ 1」が指定されていても、Ex-T スプライン補間実行中に割り込みが発生した場合は、「停止タイプ 2」と同じ減速で停止します。
(21) 付加軸は、Ex-T スプライン補間の開始位置から動作しません。途中の経路点に付加軸の位置が設定されていても、Ex-T スプライン補間では使われません。
(22) EMvSpl 命令に対して、ステップ送りをおこなうことはできますが、ステップ戻しをおこなうことはできません。エラー L2612 (ステップ戻しは実行できません) が発生します。
(23) EMvSpl 命令を実行するには、メカの制御権 (GetM 命令) が必要です。

(24) 起動条件が ALWAYS・ERROR のスロットで EMvSpl 命令を実行することはできません。エラー L2612(この起動条件では実行できません) が発生します。
(25) EMvSpl 命令の引数に設定範囲外の値が設定されている場合は、エラー L3110(引数の値が範囲外です (EMvSpl)) が発生します。

⚠注意　＜速度＞に対してあまりにも短い＜加減速距離＞を指定しますと、急な加減速となり、ロボットの振動やサーボエラーの発生につながりますので、適切な加減速距離を設定してください。

【関連命令】
SetCalFrm (セット キャリブレーションフレーム)

【関連システム状態変数】
M_SplPno、M_SplVar、P_WkCord、P_ECord

【関連パラメータ】
WK1CORD ～ WK8CORD、SPLOPTGC

End（エンド）

【機能】

プログラムの最終行を定義します。

サブルーチンをプログラムの後半に記述する場合に、メイン処理の最後に End 命令を記述してプログラムの終わりを明示的に示す場合にも使用します。CallP で呼ばれたサブプログラムの場合は End 命令実行にてメインプログラムに戻ります。

【書式】

```
End
```

【文例】

```
1  Mov P1
2  GoSub *ABC
3  End              ' プログラムの終了
   :
10  *ABC
11    M1=1
12  Return
```

【解説】

(1) プログラム最終行を定義します。プログラムを途中で停めて中断状態にしたい場合は Hlt 命令を使用してください。

(2) 操作パネルから実行した場合は連続運転のモードで実行されますので End 命令があった場合でも再度先頭から実行されます。End 命令で終了させたい場合は操作パネルの End キーを押してサイクル停止させてください。

(3) 一つのプログラム内に複数の End 命令を記述してもかまいません。

(4) プログラムの終端に End 命令の記述がなくてもかまいません。

(5) CallP で呼び出されたサブプログラムで End 命令を実行すると、メインプログラムに制御が戻ります。サブルーチン（GoSub 命令）の Return 命令と同様の動作になります。

(6) End 命令の実行で、オープンされているファイル、および、通信回線は全てクローズされます。

(7) End 命令の実行で、Spd、Accel、Oadl、JOvrd、Ovrd、Fine、Cnt の設定が初期化されます。

【関連命令】

Hlt（ホルト）、CallP（コールピー）

Error（エラー）

【機能】
プログラムにてエラー(9000 番台)を発生させます。

【書式】
通常のプログラム

```
Error □＜エラー番号＞
```

常時実行(ALWAYS)プログラム

```
Error □＜エラー番号＞,＜条件＞
```

【用語】
〈エラー番号〉　定数または数値演算式で設定できます。番号は9000～9299までの範囲で指定します。
〈条件〉　0:Error命令を実行したプログラムをエラー停止する
1:Error命令を実行したプログラムをエラー停止しない

【文例】
① エラー 9000 を発生させる。

```
1 Error  9000
```

② M1 の値によって発生させるエラーを変える

```
4 If M1 <> 0 Then *LERR          ' M1 が 0 でない場合、*LERR へ分岐する
 :
14 *LERR
15   MERR=9000+M1*10             ' M1 の値によりエラー番号を計算
16   Error MERR                  ' 計算させたエラー番号を発生させる。
17 End
```

【解説】
(1) この命令を実行することにより 9000 番台のエラーを任意に発生させることができます。
(2) ローレベル、ハイレベルエラーを発生させた場合はそのプログラムは中断します。Error 命令以降の行は実行されません。警告は中断せず、次行以降も実行します。
(3) 番号によるシステムは以下のように動作します。
(4) パラメータ UER1 ～ UER20 により、最大 20 個までエラーメッセージ等を作成することができます。
(5) エラー番号、エラーレベル、および、システムの挙動は下表の通です。
下記エラー番号の範囲以外の値を指定した場合はシステムエラーになります。

番号	システムの挙動
9000 ～ 9099 （ハイレベルエラー）	プログラムの実行を停止し、サーボ電源を遮断します。 エラーリセットの入力でエラー状態は解除されます。
9100 ～ 9199 （ローレベルエラー）	プログラムの実行を停止します。 エラーリセットの入力でエラー状態は解除されます。
9200 ～ 9299 （警告）	プログラムの実行は継続します。 エラーリセットの入力でエラー状態は解除されます。

【関連パラメータ】
UER1 ～ 20

Fine（ファイン）

【機能】

ロボットの位置決め完了条件をエンコーダパルス数で指定します。
連続動作制御中(Cnt 1)は、Fine 命令は無効になります。
Fine 命令実行時に、Fine J 命令、または Fine P 命令は全軸無効となります。

【書式】

Fine □＜パルス数＞ [, ＜軸番号＞]

【用語】

＜パルス数＞	位置決めエンコーダパルスを変数または定数で指定します。 0を指定すると無効になります。初期値は0が設定されています。
＜軸番号＞	位置決めパルスを指定する軸番号を指定します。省略すると全軸になります。定数または整数型の数値変数で指定します。

【文例】

```
1 Fine 300              ' 全軸の位置決めパルスを 300 に指定
2 Mov P1
3 Fine 100, 2           ' 2 軸の位置決めパルスを 100 に変更
4 Mov P2
5 Fine 0, 5             ' 5 軸の位置決めパルス指定を無効にする
6 Mov P3
7 Fine 0                ' 全軸の位置決めパルス指定を無効にする
8 Mov P4
```

【解説】

(1) Fine 命令は、動作命令の完了条件（位置決め精度）を、フィードバックのパルス数で指定するものです。
(2) Fine 命令が有効な全ての軸の指令パルスとフィードバックパルスの差が、＜パルス数＞以内となった時、動作完了と見なします。
(3) また、Fine 命令のかわりに Dly 命令（タイマ）で位置決めをする場合もあります。こちらの方が簡単に指定できます。

```
1 Mov P1
2 Dly 0.1
```

(4) 初期状態は、全軸 Fine 無効です。一度 Fine を有効にすると、無効にするまで有効状態が継続します。
(5) プログラム終了時(End 命令実行、中断後のプログラムリセット)に Fine は無効となります。
(6) 連続動作制御有効状態(Cnt 1)になると、Fine 有効は一時的に無視（無効、状態は保持）されます。
(7) Fine 命令実行時に、Fine J 命令、または Fine P 命令は全軸無効となります。（状態は保持されません。）
(8) 付加軸（汎用サーボ軸）に対しては、Fine 有効／無効の切換は可能ですが、パルス数の指定はできません。サーボアンプのパラメータ「INP」の設定値が、位置決めのパルス数となります。すなわち、0 以外の整数を指定した場合は、サーボアンプのパラメータ設定値で Fine が有効になり、0 を指定した場合は、Fine が無効になります。
(9) コンプライアンスモードが機能しているときに Fine 命令により位置決め完了条件を指定すると、動作によってはロボットが目的位置の位置決め完了パルスに到達することができず、動作命令の完了を待ち続けプログラムの実行が進まなくなる場合があります。コンプライアンスモードと Fine 命令を同時に使用しないでください。
(10) Fine 命令は、モータ軸のエンコーダパルスを用いて位置決め判定をおこなうため、エンコーダ分解能やギア比の影響を受けます。1 パルスあたりの出力角度は機種や軸によって異なりますので、出力軸単位(mm または度)で位置決めを監視する場合は、Fine P 命令、Fine J 命令をご使用ください。

Fine J（ファイン ジョイント）

【機能】

ロボットの位置決め完了条件を関節軸の値で指定します。
連続動作制御中(Cnt 1)は、Fine J 命令は無効になります。
Fine J 命令実行時に、Fine 命令、または Fine P 命令は全軸無効となります。

【書式】

Fine □＜位置決め幅＞, J[, ＜軸番号＞]

【用語】

〈位置決め幅〉	位置決め幅を変数または定数で指定します。 0を指定すると無効になります。初期値は0が設定されています。 単位は、関節軸の単位系により、[mm]または[deg]のどちらかとなります。 指定できる最小値は0.001です。
〈軸番号〉	位置決めパルスを指定する軸番号を指定します。省略すると全軸になります。 定数または数値変数で指定します。

【文例】

```
1 Fine 1, J            ' 全軸の位置決め幅を 1[mm]（または [deg]）に指定
2 Mov P1
3 Fine 0.5, J, 2       ' 2 軸の位置決め幅を 0.5[mm]（または [deg]）に変更
4 Mov P2
5 Fine 0, J, 5         ' 5 軸の位置決め幅指定を無効にする
6 Mov P3
7 Fine 0, J            ' 全軸の位置決め幅指定を無効にする
8 Mov P4
```

【解説】

(1) Fine J 命令は、動作命令の完了条件（位置決め精度）を、フィードバックの関節値で指定するものです。動作完了を関節値で判断するため、より正確な位置決めが可能となります。
(2) Fine J 命令が有効な全ての軸の指令関節位置とフィードバック関節位置の差が、＜位置決め幅＞以内となった時、動作完了と見なします。
(3) また、Fine J 命令のかわりに Dly 命令（タイマ）で位置決めをする場合もあります。こちらの方が簡単に指定できます。

```
1 Mov P1
2 Dly 0.1
```

(4) 初期状態は、全軸 Fine J 無効です。一度 Fine J を有効にすると、無効にするまで有効状態が継続します。
(5) プログラム終了時(End 命令実行、中断後のプログラムリセット)に Fine J は無効となります。
(6) 連続動作制御有効状態(Cnt 1)になると、Fine J 有効は一時的に無視（無効、状態は保持）されます。
(7) Fine J 命令実行時に、Fine 命令、または Fine P 命令は全軸無効となります。（状態は保持されません。）
(8) 付加軸（汎用サーボ軸）に対しても、Fine J 有効 / 無効の切換や、＜位置決め幅＞の指定は可能です。
(9) コンプライアンスモードが機能しているときに Fine J 命令により位置決め完了条件を指定すると、動作によってはロボットが目的位置の位置決め完了パルスに到達することができず、動作命令の完了を待ち続けプログラムの実行が進まなくなる場合があります。コンプライアンスモードと Fine J 命令を同時に使用しないでください。

Fine P（ファイン ポーズ）

【機能】

ロボットの位置決め完了条件を直線距離で指定します。
連続動作制御中(Cnt 1)は、Fine P 命令は無効になります。
Fine P 命令実行時に、Fine 命令、または Fine J 命令は全軸無効となります。

【書式】

Fine □＜直線距離＞, P

【用語】

〈直線距離〉　位置決め直線距離[mm]を変数または定数で指定します。
0を指定すると無効になります。初期値は0が設定されています。
指定できる最小値は0.001です。

【文例】

```
1 Fine 1, P          ' 位置決め直線距離を 1mm に指定
2 Mov P1
3 Fine 0, P          ' 位置決め直線距離指定を無効にする
4 Mov P2
```

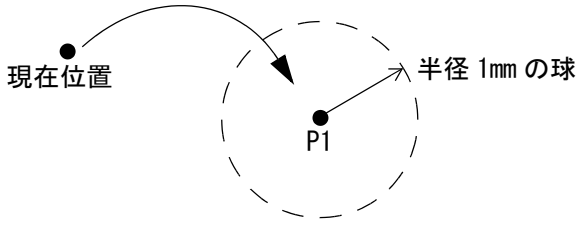
【解説】

(1) Fine P 命令は、動作命令の完了条件（位置決め精度）を、フィードバックの直線距離で指定するものです。動作完了を直線距離で判断するため、より正確な位置決めが可能となります。
(2) 指令パルスとフィードバックパルスから求められるの、それぞれのロボットの現在位置の間の直線距離が、＜直線距離＞以内となった時、動作完了と見なします。
(3) また、Fine P 命令のかわりに Dly 命令（タイマ）で位置決めをする場合もあります。こちらの方が簡単に指定できます。

```
1 Mov P1
2 Dly 0.1
```

(4) 初期状態は、全軸 Fine P 無効です。一度 Fine P を有効にすると、無効にするまで有効状態が継続します。
(5) プログラム終了時(End 命令実行、中断後のプログラムリセット)に Fine P は無効となります。
(6) 連続動作制御有効状態(Cnt 1)になると、Fine P 有効は一時的に無視（無効、状態は保持）されます。
(7) Fine P 命令実行時に、Fine 命令、または Fine J 命令は全軸無効となります。（状態は保持されません。）
(8) 付加軸（汎用サーボ軸）に対しては、Fine P 有効 / 無効の切換はできません。常に Fine P は無効となります。
(9) コンプライアンスモードが機能しているときに Fine P 命令により位置決め完了条件を指定すると、動作によってはロボットが目的位置の位置決め完了パルスに到達することができず、動作命令の完了を待ち続けプログラムの実行が進まなくなる場合があります。コンプライアンスモードと Fine P 命令を同時に使用しないでください。

```
Fine 1, P
Mov P1
```



P1 を中心とした半径 1mm の球に入ったら位置決め完了となります。

図 4-19：Fine P 命令使用例

For ～ Next（フォー ネクスト）

【機能】

For 文と Next 文の間のプログラムを、終了条件を満たすまで繰り返し実行します。

【書式】

```
For □＜カウンタ＞=＜初期値＞□ To □＜終了値＞□ [Step　＜増分＞]
  :
Next □ [＜カウンタ＞]
```

【用語】

〈カウンタ〉　繰り返し制御のカウンタとして数値変数を記述します。
〈初期値〉　繰り返し制御のカウンタの初期値を数値演算式で設定します。
〈終了値〉　繰り返し制御のカウンタの終了値を数値演算式で設定します。
〈増分〉　繰り返し制御のカウンタの増分を数値演算式で設定します。
Step以降を省略すると増分は1となります。

【文例】

① 1 から 10 までの合計を求めるプログラム

```
1 MSUM=0                    ' 合計 MSUM を初期化します。
2 For M1=1 To 10            ' 数値変数 M1 は、1 から 10 まで +1 ずつカウントアップされます。
3   MSUM=MSUM+M1            ' 数値変数 MSUM に M1 の値をたします。
4 Next M1                   ' ステップ番号 2 に戻ります。
```

②二つの数値をかけあわせた結果を 2 次元の配列変数に設定するプログラム

```
1 Dim MBOX(10,10)           ' 10×10 の配列の領域を確保します。
2 For M1=1 To 10 Step 1     ' 数値変数 M1 は、1 から 10 まで +1 ずつカウントアップされます。
3   For M2=1 To 10 Step 1   ' 数値変数 M2 は、1 から 10 まで +1 ずつカウントアップされます。
4     MBOX(M1,M2)=M1*M2     ' 配列変数 MBOX(M1,M2) に M1*M2 の値を代入します。
5   Next M2                 ' ステップ番号 3 へ戻ります。
6 Next M1                   ' ステップ番号 2 へ戻ります。
```

③ For ～ Next の中から Break で Next の次行へ抜けることができます

```
1 MSUM=0                    ' 合計 MSUM を初期化します。
2 For M1=1 To 10            ' 数値変数 M1 は、1 から 10 まで +1 ずつカウントアップされます。
3  MSUM=MSUM+M1             ' 数値変数 MSUM に M1 の値をたします。
4  If M_In(8)=1 Then Break  ' 入力信号 8 が ON していると、ステップ番号 6 へジャンプします。
5 Next                      ' ステップ番号 2 へ戻ります。
6 If M_BrkCq=1 Then Hlt
```

【解説】

(1) For 文と Next 文との間に別の For ～ Next を記述できます。ただし、一組の For ～ Next の制御は、プログラム内の制御構造を一段深くします。最高 16 段の深さまでプログラムの制御構造を深くすることができます。16 段を超えた場合は、実行時エラーとなります。
(2) For 文と Next 文との間から GoTo 命令にて強制的にジャンプすると制御構造用メモリ（スタックメモリ）が減少しますので連続で実行している場合はいつかエラーになります。For 文の条件成立にてループを抜けるようにプログラムしてください。
(3) 次の条件の場合、実行時エラーとなります。
・カウンタの＜初期値＞が＜終了値＞より大きく＜増分＞が正の値の場合
・カウンタの＜初期値＞が＜終了値＞より小さく＜増分＞が負の値の場合
(4) For 文と Next 文が対になっていない場合は、実行時エラーとなります。
(5) Next 文のカウンタ変数は、省略することが可能です。文例②では、ステップ番号 5 の M2 とステップ番号 6 の M1 を省略することができます。カウンタ変数を省略した方が、若干処理が早くなります。
(6) For ～ Next の中に Break 文があれば、Next の次行へジャンプし、For ～ Next の処理から抜けます。

FPrm（エフパラム）

【機能】

プログラムを、引数があるサブプログラム（メインプログラムから CallP で呼び出されるプログラム）として使用する場合の、引数を定義します。

【書式】

FPrm □＜仮引数＞[, ＜仮引数＞]…

【用語】

〈仮引数〉　メインプログラムからの引数の値を受け取る変数で､任意の型の変数が使用可能です。
最大16変数まで記述が可能です。

【文例】

＜メインプログラム＞

```
1 M1=1
2 P2=P_Curr
3 P3=P100
4 CallP "100",M1,P2,P3        ' 「CallP "100", 1, P_Curr, P100」とも記述できます。
```

＜サブプログラム"100"＞

```
1 FPrm M1,P1,P2
2 If M1=1 Then GoTo *LBL
3 Mov P1
4 *LBL
5 Mvs P2
6 End                         ' メインプログラムに戻ります。
```

【解説】

(1) サブプログラムの呼び出しに引数を使用しない場合は、FPrm 文は不要です。
(2) CallP の引数と、FPrm で定義された仮引数の型や個数が異なる場合はエラーとなります。
(3) サブプログラムでの処理結果を引数に代入して、メインプログラムへ渡すことはできません。
サブプログラムの処理結果をメインプログラムで使用する場合は、外部変数を使用して値を渡すようにしてください。

【関連命令】

CallP（コールピー）

GetM（ゲットメカ）

【機能】

マルチタスク使用時にスロット 1 以外のプログラムでロボット制御を行う場合、または付加軸をユーザ定義メカとしてマルチメカの制御を行う場合に、本命令を使用します。

制御するロボットのメカ番号を指定することで、制御権を獲得します。制御権を解放する場合は、RelM 命令を使用します。

【書式】

GetM □＜メカ番号＞

【用語】

〈メカ番号〉　1～3,定数または変数で記述します。
標準システムのロボットはメカ1になります。

【文例】

① タスクスロット 1 からタスクスロット 2 を起動し、タスクスロット 2 の中でメカ 1 を制御する

タスクスロット 1

```
1 RelM                  ' スロット 2 でメカ 1 を制御するために、メカを解放する。
2 XRun  2,"10"          ' スロット 2 にプログラム 10 を選択
3 Wait M_Run(2)=1       ' スロット 2 の始動確認待ち
 :
```

タスクスロット 2（プログラム "10"）

```
1 GetM 1                ' メカ 1 の制御権を獲得する。
2 Servo On              ' メカ 1 のサーボオン
3 Mov P1
4 Mvs P2
5 P3=P_Curr             ' メカ 1 の現在位置を P3 に代入
6 Servo Off             ' メカ 1 のサーボオフ
7 RelM                  ' メカ 1 の制御権を解放する
8 End
```

【解説】

(1) 通常（シングルタスク）は初期状態でメカ 1 の制御権を獲得していますので GetM は使用する必要はありません。

(2) 複数のタスクで同時に 1 つのメカの制御権を獲得することはできないため、スロット 1 以外でロボットを動作させるには、次の手順が必要です。
まず、スロット 1 のプログラムで、RelM 命令により制御権を解放します。
次に、ロボットを動作させるスロットのプログラムで、GetM 命令により制御権を獲得します。既に制御権を得ているスロットで、再度 GetM 命令を実行するとエラーとなります。

(3) 制御権が必要な命令には、モータ電源 ON/OFF 命令、補間命令、速度・加減速指定命令、Tool/Base 命令などがあります。制御権を必要とする命令の詳細は、下記を参照ください。

ロボット命令	Accel、Base、CavChk、Cmp Jnt、Cmp Pos、Cmp Tool、Cmp Off、CmpG、Cnt、ColChk、ColLvl、Def Plt、Plt、Fine、Fsc、FsGChg、HOpen、HClose、JRC、LoadSet、Mov、Mva、Mvc、Mvr、Mvr2、Mvr3、Mvs、MvSpl、MvTune、Mxt、Oadl、P_Base、Prec、Servo、SetCalFrm、Spd、SpdOpt、Tool、Torq、Trk、EMvc、EMvr、EMvr2、EMvr3、EMvs
関数	Align、PosCq、PosMid ※ 以下は、途中バージョンから GetM をしないで使用した場合でもエラーとせず、メカ 1 で使えるようにしたコマンド JtoP、PtoJ... Ver.N8/P8 ～ Inv..... Ver.R3a/S3a（特殊 ES Ver.R3dZ） Fram..... Ver.R3a/S3a（特殊 ES Ver.R3dZ）
ロボット状態変数	J_Curr、J/Fbc、J_Origin、P_Curr、P_CurrR、P_Fbc、P_Safe、P_Tool、M_Rspd、M_SetAdl

(4) メカの指定が必要なシステム状態変数の引数を省略した場合、現在取得しているメカが指定されます。

(5) プログラムを中断した場合は、システムにより自動的に RelM が実行され、再始動時には自動的に GetM が実行されます。
(6) 常時実行プログラム内では使用できません。

【関連命令】
RelM（リリースメカ）

GoSub(～Return)（ゴーサブ リターン）

【機能】
指定されたラベルのサブルーチンを呼び出します。
サブルーチンからは必ず Return 命令で戻します。

【書式】

```
GoSub □＜呼び出し先＞
```

【用語】
〈呼び出し先〉　　ラベル名を記述します。

【文例】

```
 10  GoSub *LBL
 11  End
   :
100 *LBL
101 Mov P1
102 Return              ’ かならず Return 命令で戻す。
```

【解説】
(1) サブルーチンからは、必ず Return 命令で戻すようにしてください。GoTo 命令にて戻すと制御構造用メモリ（スタックメモリ）が減少し、連続運転時のエラー発生の原因となります。
(2) サブルーチンの中から、GoSub により再度他のサブルーチンの呼び出しが可能です。サブルーチン呼び出し可能段数は、約 800 段となります。
(3) 呼び出し先は、ラベルで指定します。呼び出し先のラベルが存在しない場合、実行時エラーとなります。

【関連命令】
Return（リターン）

GoTo(ゴートゥー)

【機能】
プログラム内で指定されたラベルへ無条件に分岐します。

【書式】

```
GoTo □<分岐先>
```

【用語】
〈分岐先〉　　ラベル名を記述します。

【文例】

```
 10  GoTo *LBL  ' ラベル *LBL へ分岐します。
   :
100 *LBL
101 Mov P1
```

【解説】
(1) 分岐先にはラベルを指定します。
(2) 分岐先のラベルが存在しない場合は実行時エラーになります。

Hlt(ホルト)

【機能】
本 Hlt 命令を実行したプログラムを中断し、待機中とします。
マルチタスク機能をご利用の場合、他のプログラムの実行状態には影響しません。

【書式】

```
Hlt
```

【文例】
① ある条件を満たした時、ロボットを停止させる。

```
100 If M_In(18)=1 Then Hlt          ' 入力信号 18 が ON した場合はプログラムを中断します。

200 Mov P1 WthIf M_In(17)=1, Hlt    ' P1 へ移動中に、入力信号 17 が ON した場合はプログラムを中
                                      断します。
```

② 無条件にプログラムの途中でロボットを停止させる。

```
150 Hlt                             ' 無条件にプログラムを中断します。
```

【解説】
(1) 本 Hlt 命令を実行したプログラムを中断し、待機中とします。
(2) マルチタスク機能をご利用の場合、本命令を実行したタスクスロットのみ実行を中断します。
(3) 再開は、O/P の始動または外部からの始動信号で行ないます。プログラムは Hlt 文の次の行から実行再開します。ただし、付随節(WthIf 命令)の Hlt の場合は実行中断した行から再度実行されます。

【関連命令】
End(エンド)

⚠注意　トラッキング機能をご利用の場合
トラッキング動作中に本 Hlt 命令を実行すると、トラッキング動作を停止し(Trk Off 命令相当)プログラムの実行を中断します。マルチメカをご使用の場合は、GetM 命令で取得しているメカ番号のロボットに対してトラッキング動作を停止します。
再起動(継続)でトラッキング動作を継続する場合は、Trk On 命令を実行するようプログラムを作成してください。

HOpen/HClose（ハンドオープン／ハンドクローズ）

【機能】
ハンドの開閉を指令します。
【書式】

HOpen □＜ハンド番号＞ [, ＜起動把持力＞, ＜保持把持力＞, ＜起動把持力継続時間＞]

HClose □＜ハンド番号＞

【用語】

＜ハンド番号＞	1～8の数値のいずれかを選択します。定数または変数で指定します。
＜起動把持力＞注)	電動ハンドの場合に有効なパラメータです。電動ハンド以外の場合は無効。 ハンド開閉を起動するために必要な把持力を定数または変数で設定します。 0から63段階(63=3.5kg)で把持力を設定します。 初期値は63です。省略時は前回設定値が有効です。
＜保持把持力＞注)	電動ハンドの場合に有効なパラメータです。電動ハンド以外の場合は無効。 ハンド開閉を保持するために必要な把持力を定数または変数で設定します。 0から63段階(63=3.5kg)で把持力を設定します。 初期値は63です。省略時は前回設定値が有効です。
＜起動把持力継続タイマー＞注)	電動ハンドの場合に有効なパラメータです。 起動把持力を継続している時間を定数または変数で設定します。 0.00(秒)から精度実数の最大値まで設定できます。 初期値は0.3秒です。

注）弊社製電動ハンドのみに有効です。
TAIYO 社製多機能電動ハンドの制御については 547 ページの「7.2 多機能電動ハンドの使用方法」を参照願います。

【文例】

```
1 HOpen 1                ' ﾊﾝﾄﾞ 1 を開く。
2 Dly 0.2                ' ﾀｲﾏを 0.2 秒 （ハンド開完了待ち）
3 HClose 1               ' ﾊﾝﾄﾞ 1 を閉じる。
4 Dly 0.2                ' ﾀｲﾏを 0.2 秒 （ハンド閉完了待ち）
5 Mov PUP                '
```

【解説】
(1) 各ハンドの動作の設定（シングルソレノイド / ダブルソレノイド）はパラメータ HANDTYPE で設定します。
(2) ダブルソレノイドに設定した場合はハンドは 1 ～ 4 まで対応可能です、シングルソレノイドに設定した場合はハンドは 1 ～ 8 まで対応します。
(3) 電源投入時のハンド出力信号の状態は、パラメータ HANDINIT で設定します。
(4) ハンド入力信号の確認は、ロボット状態変数 M_HndCq(’ハンド入力番号’)で行えます。また、900 ～ 907 番の入力信号（メカが 1 台の場合）でも確認が可能です。

```
1 HClose 1
2 *LBL: If M_HndCq(1)<>1 Then GoTo *LBL
3 Mov P1
```

(5) 関連する項目にパラメータがあります。本書 /442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」および 445 ページの「5.13 ハンド初期状態について」を参照してください。

【関連システム状態変数】
M_In(900 番台）、M_Out(900 番台）、M_HndCq
【関連命令】
LoadSet（ロードセット）、Oadl（オーエーディーエル）
【関連パラメータ】
HANDTYPE、HANDINIT
442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」および、445 ページの「5.13 ハンド初期状態について」を参照してください。

If…Then…Else…EndIf(イフ ゼン エルス)

【機能】

式の結果により、処理を選択し実行します。

【書式】

```
If □ < 式 > □ Then □ < 処理 > □ [Else  < 処理 >]
```

```
If □ < 式 > □ Then
  < 処理 >
  < 処理 >
  Break
    :
[Else
  < 処理 >
  < 処理 >
  Break
    : ]
EndIf
```

【用語】

〈式〉　比較の対象となる式を比較演算式､または論理演算式で記述します。

〈処 理〉　Thenに続く処理に比較の結果が真の時の処理を､Elseに続く処理に偽の時の処理を記述します。

【文例】

① If…Then…Else… を 1 行で表記するとき

```
10 If M1>10 Then *L100                      ' M1 が 10 より大きい場合はラベル L100 へジャンプ
11 If M1>10 Then GoTo *L20 Else GoTo *L30 ' M1 が 10 より大きい場合は ラベル L20 へ、
                                              10 以下の場合はラベル L30 へジャンプ
                                              Then、Else の後の GoTo は省略できます。
```

② If…Then…Else…EndIf のブロック構造を用いたとき

```
10 If M1>10 Then
11   M1=10
12   Mov P1
13 Else
14   M1=-10
15   Mov P2
16 EndIf
```

③ Then または Else の中に、さらに If…Then…Else…EndIf のブロックを記述した場合

```
30 If M1>10 Then
31   If M2 > 20 Then
32     M1 = 10
33     M2 = 10
34   Else
35     M1 = 0
36     M2 = 0
37   EndIf
38 Else
39   M1 = -10
400   M2 = -10
410 EndIf
```

④ Then または Else の中で Break で EndIf の次行へ抜けられます。

```
30 If M1>10 Then
31   If M2 > 20 Then Break   ’条件が成立したらステップ番号 39 へジャンプします。
32   M1 = 10
33   M2 = 10
34 Else
35   M1 = -10
36   If M2 > 20 Then Break   ’条件が成立したらステップ番号 39 へジャンプします。
37   M2 = -10
38 EndIf
39 If M_BrkCq=1 Then Hlt
40 Mov P1
```

【解説】

(1) If‥Then‥Else‥は一行で記述します。
(2) If‥Then‥Else‥EndIf は複数行にまたがって記述できます。
(3) Else は省略できます。
(4) If‥Then‥Else‥EndIf の場合は EndIf は必ず記述してください。
(5) If‥Then‥Else‥EndIf の条件分岐処理の中で GoTo 命令にてジャンプさせると制御構造用メモリ（スタックメモリ）がたりなくなりエラーになります。
(6) If‥Then‥Else‥EndIf の場合、Then または Else の中に If‥Then‥Else‥EndIf を記述することが可能です。（ネスティングは 8 段まで可能）
(7) Then または、Else の後に GoTo が続く場合、GoTo を省略することが可能です。

例）If M1 > 10 Then *L200 Else *L300

また、Then の後に GoTo が続く場合のみ、Then または GoTo のいずれかを省略することが可能です。Else は省略できません。

例） If M1 > 10 Then GoTo *L200 （左記プログラムは下の様に書換えることができます）
⇒ If M1 > 10 Then *L200
⇒ If M1 > 10 GoTo *L200

(8) If Then EndIf の中で Break 文があれば EndIf の次行へジャンプし If Then EndIf の処理から抜けます。
(9) 条件式には、論理数を記述することができます。論理数は 0 でなければ真（条件成立）、0 で偽を表します。したがって次のような記述も可能です。

例）If M_IN(900) Then M_Out(30)=1

入力信号 900 が ON の場合は真(M_In(900)=1)として処理されるので、Then 以降の記述が実行されます。
入力信号 900 が OFF の場合は偽(M_In(900)=0)として処理されるので、Then 以降の記述は実行されません。

Input（インプット）

【機能】

ファイル（通信回線を含む）からデータを入力します。受け取れるデータは全て ASCII 文字です。

【書式】

```
Input □ #< ファイル番号 >, < 入力データ名 >[, < 入力データ名 >]…
```

【用語】

〈ファイル番号〉　1～8までの番号を記述します。
Open命令で割り当てたファイル番号に対応します。

〈入力データ名〉　入力したデータを取り込む変数名を記述します。すべての変数を記述できます。

【文例】

```
1 Open "temp.txt" For Input As #1  ' ファイル番号 1 に "temp.txt" を割り当てます。
2 Input #1, CABC$       ' ファイル番号 1 から改行コードまでの文字列を CABC$ に読み込みます。
 :
10 Close #1             ' ファイル番号 1 を閉じます。
```

【解説】

(1) Open 文で開かれているファイル番号のファイルからデータを入力し、変数に代入します。Open 文が実行されていない場合はエラーになります。
(2) 入力されるデータと、それを代入する変数の型は同じである必要があります。
(3) 複数の変数名を記述する場合は、変数名の間をカンマ(,)で区切って記述します。
(4) Input 文が実行されると、入力待ちの状態になります。入力したデータは、改行(CR, LF)コードの入力と同時に、変数に代入されます。
(5) Input 文の引数の数より、多い要素数が入力された場合は、読み捨てられます。End または Close 実行時は、バッファに溜まっているデータは消滅します。

例）文字列と数値と位置を混在させて入力する場合

```
1 Input #1,C1$,M1,P1
```

パソコン側から送るデータ例

（ロボットの標準ポートにて受け取る場合：パラメータ "CPRC232" が 0 の場合）

```
PRN MELFA,125.75,(130.5,-117.2,55.1,16.2,0,0)(1,0) CR
```

C1$ に MELFA、M1 に 125.75、P1 に(130.5,-117.2,55.1,16.2,0,0)(1,0) が、代入されます。

【関連命令】

Open（オープン）、Close（クローズ）、Print（プリント）

JOvrd（ジェーオーバーライド）

【機能】
ロボットの関節移動時にのみ有効なオーバーライドを指定します。

【書式】

JOvrd □＜指定オーバーライド＞

【用語】
〈指定オーバーライド〉　オーバーライドを実数で指定します。数値演算式でも記述できます。
単位:[%](推奨範囲:0.1～100.0)

【文例】
```
1 JOvrd 50
2 Mov P1
3 JOvrd M_NJOvrd        ' 初期値を設定
```

【解説】
(1) JOvrd 命令は関節補間時のみに有効です。
(2) 実際のオーバーライドは =(操作パネル(T/B)のオーバーライド設定値)×(プログラムオーバーライド(Ovrd 命令))×(関節オーバーライド(JOvrd 命令))です。JOvrd 命令は関節補間動作のオーバーライドのみを変化させます。
(3)＜指定オーバーライド＞は、100% がロボットの能力の最大となります。通常、システム初期値(M_NOvrd)は 100% が設定されています。End 文の実行、またはプログラムリセットで初期値に戻ります。

【関連命令】
Ovrd（オーバーライド）、Spd（スピード）

【関連システム状態変数】
M_NJOvrd(システムの初期値)、M_JOvrd(現在の指定関節オーバーラード)、

JRC（ジョイント・ロール・チェンジ）

【機能】

・ロボット本体の対象軸（【用語】の＜軸番号＞参照）の現在の関節座標値に、±360 度を加算して現在座標値を書き換えます。

・ユーザ定義軸（付加軸、ユーザ定義メカ）
指定した軸の現在の関節座標値に、パラメータで指定した値を加減算して現在座標値を書き換えます。回転軸、直動軸のいずれでも使用可能です。現在座標値での再原点設定も可能です。

【書式】

JRC □ <[+]＜数値＞/-＜数値＞/0> [, ＜軸番号＞]

【用語】

〈数値〉　増加減分の数(360度の倍数)を指定します。定数または変数での記述が可能です。
例)+3:対象軸の角度を+1080度増加させます。 -2:-720度減少させます。
+1/-1/0を指定した場合は、次のような意味を持ちます。
+1: 指定軸の現在の関節角度をパラメータJRCQTTで指定した分増加させる。(符号省略可能)
ロボット本体の優先軸の場合は、360度固定です。
-1: 指定軸の現在の関節角度をパラメータJRCQTTで指定した分減少させる。
ロボット本体の優先軸の場合は、360度固定です。
0: 指定軸をパラメータJRCORGで指定した値で再原点設定する。
ユーザ定義軸のみ使用可能。

〈軸番号〉　対象とする軸を定数または変数で指定します。省略時は優先軸になります。
なお、付加軸、ユーザ定義メカの軸を対象とする場合は省略できません。
[対象機種と対象軸]
(1)ロボット本体

RH-F シリーズ：J4 軸（優先軸）
RV-F シリーズ：J6 軸（優先軸）

(2)全機種の付加軸
(3)ユーザ定義メカの全軸

【文例】

```
1 Mov P1        ’P1 へ移動します。(J6 軸がマイナス方向へ動作する動き）
2 JRC +1        ’ 対象軸の現在座標値を 360° 加算します。
3 Mov P1        ’P1 へ移動します。
4 JRC +1        ’ 対象軸の現在座標値を 360° 加算します。
5 Mov P1        ’P1 へ移動します。
6 JRC -2        ’ 対象軸の現在座標値を -720° 減算します。（元に戻す）
```

【解説】

(1) JRC 1/-1 命令(JRC n/-n)により、指定した軸の現在の関節座標値が指定した分増減します。
JRC 0 命令により、指定した軸を再原点設定します。
関節座標の値が変化しますが、ロボットは動きません。
(2) 本命令を使用する場合には、命令実行により動作範囲外とならないよう、あらかじめ対象軸の動作範囲を変更してください。変更は、関節動作範囲パラメータ"MEJAR"の該当する軸の－側、+側の値を変更することで可能です。なお回転軸の動作範囲は、-2340 ～ +2340 度の範囲で設定してください。
(3) 指定軸を省略したときは、優先軸が対象軸となります。優先軸はロボット先端の回転軸となります。
(4) 優先軸が存在しない(JRC 不可能なロボット）場合に指定軸を省略したり、指定した軸が JRC 非対象軸のときは、実行時エラーとなります。
(5) 原点設定未のときは、実行時エラーとなります。
(6) JRC 命令実行時、ロボットは停止します。Cnt を有効にしている場合でも、この命令を実行すると補間の接続は連続動作となりません。

(7) JRC 命令を使用するときは、下記パラメータの設定が必要となります。
JRCEXE = 1 とします。(JRC 実行可能)
MEJAR で、対象軸の動作範囲を変更します。
JRCQTT で、JRC 1/-1(JRC n/-n) 実行時の位置変化量を設定します。
(付加軸、ユーザ定義メカの場合のみ)
JRCORG で、JRC 0 実行時の原点位置を設定します。(付加軸、ユーザ定義メカの場合のみ)
(8) パラメータ JRCEXE = 0 のときは、JRC 命令を実行しても無処理となります。
(9) パラメータ JRCQTT で指定した移動量がパルスデータで 0 ～ 2147483647(0x7fffffff) に収まらない場合は初期化時エラーとなります。
また、移動量 → パルスデータの換算は、
回転軸の場合
パルスデータ = 移動量 (deg) / 360 * ギヤ比分母 / ギヤ比分子 * エンコーダパルス数
直動軸の場合
パルスデータ = 移動量 (mm) * ギヤ比分母 / ギヤ比分子 * エンコーダパルス数
となります。
(10) JRC の実行により、原点データが変わりますので、出荷時の原点データは使用できなくなります。バージョンアップ作業等で、コントローラの初期化が必要となる場合は、あらかじめ直前の状態での、パラメータバックアップが必要となります。
(11) JRC 命令はステップ戻し操作はできません。
(12) 常時実行プログラム内では使用できません。
(13) 本コマンドを使用して、比較的大きな移動量を伴う動作を連続的におこなった場合、「過負荷エラー」や「サーボアンプモータオーバーヒートエラー」が発生することがあります。この場合は、速度や加減速度割合を調節してエラーが発生しない条件でご使用ください。

【関連パラメータ】
JRCEXE
JRCの実行可/不可を設定します。
実行不可=0(初期値)/実行可=1
JRCQTT
付加軸､ユーザー定義メカにおいて､JRC命令で増減させる変化量(1deg/1mm単位)を指定します。
ロボット本体側のJRC対象軸の場合はこの設定によらず､360度に固定です。
JRCORG (JRC 0実行時の原点/float型/要素数8/ユーザパラメータ)
付加軸､ユーザー定義メカにおいて､JRC 0を実行したときの原点座標値を指定します。

各パラメータの詳細は､408ページの｢5 パラメータでの設定機能｣を参照願います。

LoadSet（ロードセット）

【機能】

Oadl 命令実行時のハンド・ワークの条件を指定します。
また、干渉回避機能使用時に、ハンド番号、ワーク番号を指定します。（干渉チェックの対象とするモデルの指定）

【書式】

LoadSet □＜ハンド条件番号＞, ＜ワーク条件番号＞

【用語】

＜ハンド条件番号＞	0～8。 最適加減速に使用する質量･大きさを指定したハンド条件パラメータ(HNDDAT0～8)の番号を定数または変数で指定します。 同時に干渉回避機能に使用するハンドモデルも変更されます。パラメータCAVKDH1～8の第1要素の番号に対応します。
＜ワーク条件番号＞	0～8。 最適加減速に使用する質量･大きさを指定したワーク条件パラメータ(WRKDAT0～8)の番号を定数または変数で指定します。 同時に干渉回避機能に使用するワークモデルも変更されます。パラメータCAVKDW1～8の第1要素の番号に対応します。

【文例】

```
1 Oadl ON
2 LoadSet 1,1      ' ハンド 1(HNDDAT1) ・ワーク 1(WRKDAT1) の条件
3 Mov P1
4 LoadSet 0,0      ' ハンド 0(HNDDAT0) ・ワーク 0(WRKDAT0) の条件
5 Mov P2
6 Oadl Off
```

【解説】

(1) 最適加減速に使用するハンドの条件、ワークの条件を設定します。重量の異なるワーク品種毎に最適な加減速を設定する場合などに使用します。また、干渉回避機能に使用するハンド、ワーク番号にも連動しています。
(2) プログラム実行開始時にはハンドに最大負荷が設定された状態になっています。
また、干渉回避機能に使用するハンド、ワークのモデルは初期値に設定されているモデルが対象になっています。(CAVKDH1 ～ 8（ハンドモデル)、CAVKDW1 ～ 8(ワークモデル) それぞれの第 1 要素に "0" が設定されているモデル)
(3) ハンド条件はパラメータ (HNDDAT0 ～ 8) に、重量、大きさ (X,Y,Z)、重心位置 (X,Y,Z) を設定します。
(4) ワーク条件はパラメータ (WRKDAT0 ～ 8) に、重量、大きさ (X,Y,Z)、重心位置 (X,Y,Z) を設定します。
(5) 本命令を実行して変更したハンド条件とワーク条件はプログラムリセット時、および END 文実行時にシステム初期値に再設定されます。
システム初期値は、HNDDATO、WRKDATO を使用します。
(6) 最適加減速用の詳細な設定は、456 ページの「5.16 ハンド、ワーク条件設定 (最適加減速設定) について」を参照してください。
干渉回避機能の詳細な設定は 481 ページの「5.24 干渉回避機能について」を参照してください。

【関連命令】

Oadl (オーエーディーエル)、HOpen/HClose (ハンドオープン / ハンドクローズ)

【関連パラメータ】

HNDDAT0 ～ 8、WRKDAT0 ～ 8、HNDHOLD0 ～ 8 については、456 ページの「5.16 ハンド、ワーク条件設定 (最適加減速設定) について」を参照してください。
ACCMODE については、408 ページの「表 5-1 : 動作関連パラメーター覧表」を参照してください。
干渉回避機能に関連するパラメータについては、481 ページの「5.24 干渉回避機能について」を参照してください。

Mov（ムーブ）

【機能】

現在位置から移動目的位置まで関節補間動作をおこないます。

【書式】

Mov □＜移動目的位置＞ [, ＜近接距離＞]

[□ Type □＜定数 1>, ＜定数 2>] □ [＜付随条件＞]

【用語】

＜移動目的位置＞	補間動作の最終位置。位置型の変数と定数、または関節変数で記述します。
＜近接距離＞	この値を指定した場合、実際の移動目的位置は、ツール座標系のZ軸方向(+/-方向)に指定した距離だけ離れた位置へ動作します。定数または変数で指定します。
＜定数1＞	1/0:遠回り/近回り動作。 初期値は1(遠回り)です。定数のみ。
＜定数2＞	無効(0を記述してください)、定数のみ。
＜付随条件＞	Wth、WthIF節が使用できます。

【文例】

```
1 Mov P1 Type 1,0
2 Mov J1
3 Mov (Plt 1,10),100.0 Wth M_Out(17)=1
4 Mov P4+P5,50.0 Type 0,0 WthIf M_In(18)=1,M_Out(20)
```

【解説】

(1) 始点位置と終点位置の各軸の関節角度差を均等に補間します。よって先端の軌跡は保証されません。
(2) Wth、WthIf 文を併用することにより、信号出力タイミングと動作の同期をとることができます。
(3) Type の数値定数 1 は姿勢の補間方式を指定します。
(4) 関節補間でいう遠回りとは、ティーチング姿勢通りに動作をするという意味です。ティーチング時の姿勢によっては近回り動作になる場合があります。
(5) 近回り動作とは、始点・終点間の姿勢において動作量が少ない方向に姿勢の補間を行ないます。
(6) 遠回り / 近回りの指定は、開始位置と目的位置の動作範囲が ±180 度以上の移動量を持っている場合に意味を持ちます。
(7) 近回りを指定した場合でも、目的位置が動作範囲外になる場合は、逆方向に遠回りをして動作します。
(8) 関節補間では、Type の数値定数 2 は意味を持ちません。
(9) 常時実行プログラム内では使用できません。
(10) Mov 命令実行中に中断してジョグ送りしたあとに再開した場合は中断した位置まで戻ってから MoV 命令を再開します。中断位置へ戻る補間方法 (関節補間 / 直線補間) は RETPATH パラメータで変更できます。また、この RETPATH パラメータの値を変更することにより、中断位置へ戻らずに直接目的位置へ移動することも可能です。(442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください)。

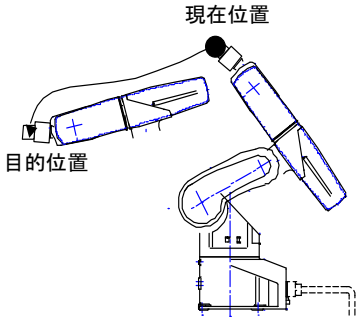


図 4-20 : 関節補間の動作軌跡例

Mva（ムーブアーチ）

【機能】

現在位置からアーチモーション動作（アーチ補間）で目的位置まで移動します。

【書式】

Mva □＜移動目的位置＞ [, ＜アーチ番号＞]

【用語】

＜移動目的位置＞　補間動作の最終位置。位置型の変数と定数、または関節変数で記述します。
＜アーチ番号＞　Def Arch(1～4)。省略時は1です。

【文例】

```
1 Def Arch 1,5,5,20,20          ' アーチ形状を定義
2 Ovrd 100,20,20                ' オーバーライドを指定
3 Accel 100,100,50,50,50,50     ' 加減速比率を設定
4 Mov P0                        ' アーチモーション動作の開始位置へ移動します。
5 Mva P1,1                      ' ステップ番号 1 で定義した形状でアーチモーション動作をします。
6 Mva P2,2                      ' パラメータの初期値で動作します。
```

【解説】

(1) 現在位置から Z 軸方向に上昇し、続いて目的位置の上空まで移動し、最後に下降して目的位置まで動作します。いわゆるアーチモーション動作を一つの命令で実行します。
(2) Def Arch 命令なしで Mva 命令を実行した場合はパラメータに設定したアーチ形状で動作します。パラメータに関しては 193 ページの「Def Arch（デフ アーチ）」を参照してください。
(3) 補間形式、補間タイプなども Def Arch 命令で定義しますので、193 ページの「Def Arch（デフ アーチ）」を参照してください。
(4) 常時実行プログラム内では使用できません。
(5) Mva 命令実行中に中断してジョグ送りしたあとに再開した場合は中断した位置まで戻ってから Mva 命令を再開します。中断位置へ戻る補間方法（関節補間 / 直線補間）は RETPATH パラメータで変更できます（442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください）。

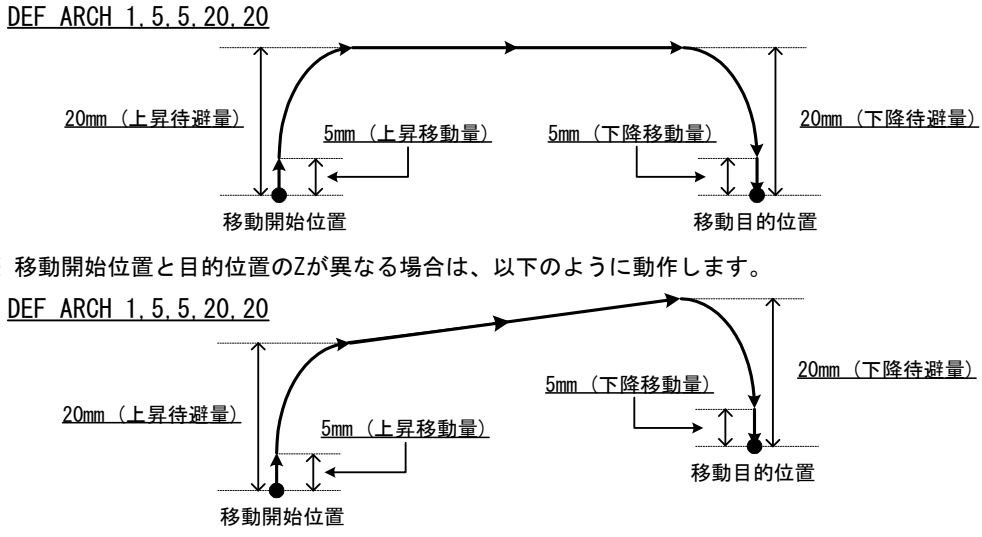


図 4-21：アーチ補間の動作軌跡例（横から見た場合）

⚠注意　ロボットは、その時の指定速度によって異なる軌跡で動作する場合があります。
特に、コーナー部分は内回り距離が変化する場合がありますので、自動運転を始める時は最初、低速で動作させ、周辺装置との干渉に注意しながら徐々に速度を上げてください。

【関連命令】
Def Arch (デフ アーチ)、Accel (アクセル)、Ovrd (オーバーライド)

Mvc（ムーブシー）

【機能】

始点、通過点 1、通過点 2、始点の順で三次元円補間動作をおこないます。

【書式】

Mvc □＜始点＞, ＜通過点 1>, ＜通過点 2> □[＜付随条件＞]

【用語】

〈始点〉　円の始点および終点。位置型の変数と定数、または関節変数で記述します。
〈通過点1〉　円弧の通過点1。位置型または関節型の定数および変数で記述します。
〈通過点2〉　円弧の通過点2。位置型または関節型の定数および変数で記述します。
〈付随条件〉　Wth節またはWthIf節を記述します。

【文例】

```
1 Mvc P1,P2,P3
2 Mvc P1,J2,P3
3 Mvc P1,P2,P3 Wth M_Out(17)=1
4 Mvc P3,(Plt 1,5),P4 WthIf M_In(20)=1,M_Out(21)=1
```

【解説】

(1) 円補間動作は、与えられた 3 点から円を求め、円周を移動します。(360 度)
(2) 円補間中の姿勢は、始点の姿勢を維持します。通過点 1, 2 の姿勢は考慮されません。
(3) 現在位置と始点が一致しない場合は、自動的に始点まで直線補間 (三軸直交補間) で移動しその後円補間をおこないます。
(4) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで関節補間で移動した後、残りの円弧補間を再開します。中断位置へ戻る補間方法 (関節補間 / 直線補間) は RETPATH パラメータで変更できます (442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください)。
(5) 常時実行プログラム内では使用できません。

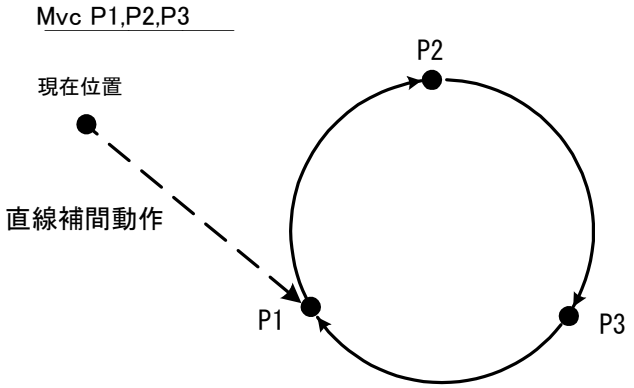


図 4-22 : 円補間の動作軌跡例

Mvr（ムーブアール）

【機能】

始点から通過点を経由して終点まで三次元円弧補間動作を行ないます。

【書式】

Mvr □＜始点＞, ＜通過点＞, ＜終点＞

[□ Type □＜定数 1>, ＜定数 2>] □ [＜付随条件＞]

【用語】

〈始点〉	円弧の始点。位置型の変数と定数、または関節変数で記述します。
〈通過点〉	円弧の通過点。位置型の変数と定数、または関節変数で記述します。
〈終点〉	円弧の終点。位置型の変数と定数、または関節変数で記述します。
〈定数1〉	近回り/遠回り動作=0/1、初期値は0。
〈定数2〉	等価回転/三軸直交/特異点通過=0/1/2、初期値は0。
〈付随条件〉	Wth節またはWthIf節を記述します。

【文例】

```
1 Mvr P1,P2,P3
2 Mvr P1,J2,P3
3 Mvr P1,P2,P3 Wth M_Out(17)=1
4 Mvr P3,(Plt 1,5),P4 WthIf M_In(20)=1,M_Out(21)=1
```

【解説】

(1) 円弧補間動作は、与えられた 3 点から円弧を求め、その円弧上を移動します。
(2) 姿勢は、始点から終点への補間となり、通過点の姿勢は影響しません。
(3) 現在位置と始点が一致しない場合は、自動的に始点まで直線補間（三軸直交補間）します。
(4) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで関節補間で移動した後、残りの円弧補間を再開します。中断位置へ戻る補間方法（関節補間 / 直線補間）は RETPATH パラメータで変更できます(442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください)。
(5) 等価回転（定数 2=0）の場合、始点と終点の構造フラグが異なる場合は、実行時にエラーとなります。
(6) 指定した 3 点のうち同一位置がある場合や、3 点が一直線上にある場合は始点から終点に向かって直線補間動作をします。エラーとはなりません。
(7) 定数 2 に三軸直交を指定した場合、定数 1 は無効でティーチング通りの姿勢に移動します。
(8) 定数 2 は姿勢補間のタイプを指定します。三軸直交とは、(X,Y,Z,J4,J5,J6) の座標系で補間を行う為、特異点近傍を通過する場合に効果的です。特異点通過は対応している機種が限定されています。詳細については 467 ページの「5.20 特異点通過機能について」を参照してください。
(9) 常時実行プログラム内では使用できません。

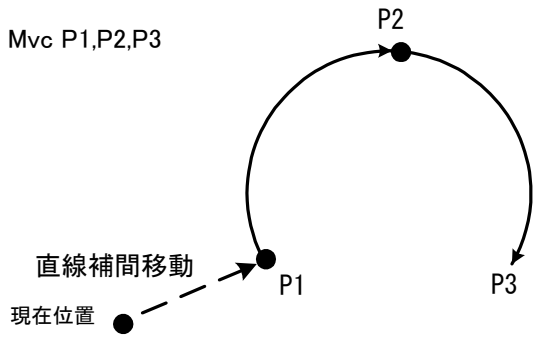


図 4-23：円弧補間の動作軌跡例その 1

◇◆◇垂直 5 軸ロボットにおいて直線(円弧)補間実行後の位置データ登録時の注意点◇◆◇

直線(円弧)補間において、動作開始位置の姿勢データと目的位置の姿勢データが、+180 度もしくは -180 度以上異なるとロボットは近回りで動作します。このとき垂直 5 軸ロボット(RV-4FJL)では、目的位置へ到達すると姿勢データは登録された値を表示していますが、実際には +360 度もしくは -360 度ずれた姿勢となっています。

例えば、A 軸の値が 0 度の位置から +200 度の位置へ直線補間させると、現在位置の姿勢データは +200 度と表示しますが、ロボットは -160 度の位置に動作しています。

この状態で位置データを登録しますと、実際とは +360 度もしくは -360 度異なった姿勢データが登録されます。プログラムのデバッグ操作(ステップ送り、MS ポジション移動)などで直線(円弧)補間動作後に位置データを補正する場合は、ご注意願います。

なお、この現象は直線(円弧)補間命令の引数 Type で遠回り動作を指定していただくか、位置データ登録時に一旦サーボオフしていただくことにより回避することが可能です。

Mvr2（ムーブアール 2）

【機能】

始点、終点、参考点から構成される円弧上を、始点から終点まで三次元円弧補間動作を行ないます。
移動方向は参考点を通過しない方向となります。

【書式】

Mvr2 □＜始点＞, ＜終点＞, ＜参考点＞
[□ Type □＜定数 1>, ＜定数 2>] □ [＜付随条件＞]

【用語】

＜始点＞	円弧の始点。位置型の変数と定数、または関節変数で記述します。
＜終点＞	円弧の終点。位置型の変数と定数、または関節変数で記述します。
＜参考点＞	円弧の参考点。位置型の変数と定数、または関節変数で記述します。
＜定数1＞	近回り/遠回り動作=0/1、初期値は0。
＜定数2＞	等価回転/三軸直交/特異点通過=0/1/2、初期値は0。
＜付随条件＞	Wth節またはWthIf節を記述します。

【文例】

```
1 Mvr2 P1,P2,P3
2 Mvr2 P1,J2,P3
3 Mvr2 P1,P2,P3 Wth M_Out(17)=1
4 Mvr2 P3,(Plt 1,5),P4 WthIf M_In(20)=1,M_Out(21)=1
```

【解説】

(1) 円弧補間動作は、与えられた 3 点から円弧を求め、その円弧上を移動します。
(2) 姿勢は、始点から終点への補間となり、参考点の姿勢は影響しません。
(3) 現在位置と始点が一致しない場合は、自動的に始点まで直線補間（三軸直交補間）します。
(4) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで関節補間で移動した後、残りの円弧補間を再開します。中断位置へ戻る補間方法（関節補間 / 直線補間）は RETPATH パラメータで変更できます（442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください）。
(5) 移動方向は参考点を通らない方向です。
(6) 等価回転（定数 2=0）の場合、始点と終点の構造フラグが異なる場合は、実行時にエラーとなります。
(7) 指定した 3 点のうち同一位置がある場合や、3 点が一直線上にある場合は始点から終点に向かって直線補間動作をします。エラーとはなりません。
(8) 定数 2 に三軸直交を指定した場合、定数 1 は無効でティーチング通りの姿勢に移動します。
(9) 定数 2 は姿勢補間のタイプを指定します。三軸直交とは、(X,Y,Z,J4,J5,J6) の座標系で補間を行う為、特異点近傍を通過する場合に効果的です。特異点通過は対応している機種が限定されています。詳細については 467 ページの「5.20 特異点通過機能について」を参照してください。
(10) 常時実行プログラム内では使用できません。

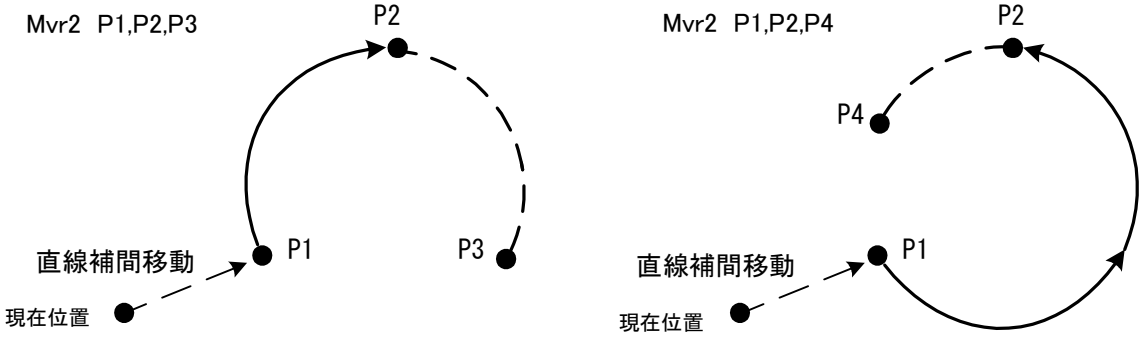


図 4-24：円弧補間の動作軌跡例その 2

◇◆◇垂直 5 軸ロボットにおいて直線(円弧)補間実行後の位置データ登録時の注意点◇◆◇

直線(円弧)補間において、動作開始位置の姿勢データと目的位置の姿勢データが、+180 度もしくは -180 度以上異なるとロボットは近回りで動作します。このとき垂直 5 軸ロボット(RV-4FJL)では、目的位置へ到達すると姿勢データは登録された値を表示していますが、実際には +360 度もしくは -360 度ずれた姿勢となっています。

例えば、A 軸の値が 0 度の位置から +200 度の位置へ直線補間させると、現在位置の姿勢データは +200 度と表示しますが、ロボットは -160 度の位置に動作しています。

この状態で位置データを登録しますと、実際とは +360 度もしくは -360 度異なった姿勢データが登録されます。プログラムのデバッグ操作(ステップ送り、MS ポジション移動)などで直線(円弧)補間動作後に位置データを補正する場合は、ご注意願います。

なお、この現象は直線(円弧)補間命令の引数 Type で遠回り動作を指定していただくか、位置データ登録時に一旦サーボオフしていただくことにより回避することが可能です。

Mvr3 (ムーブアール 3)

【機能】

始点、終点、中心点から構成される円弧上を、始点から終点まで三次元円弧補間動作を行ないます。

【書式】

Mvr3 □ < 始点 >, < 終点 >, < 中心点 >

[□ Type □ < 定数 1>, < 定数 2>] □ [< 付随条件 >]

【用語】

<始点>	円弧の始点。位置型の変数と定数、または関節変数で記述します。
<終点>	円弧の終点。位置型の変数と定数、または関節変数で記述します。
<中心点>	円弧の中心点。位置型の変数と定数、または関節変数で記述します。
<定数1>	近回り/遠回り動作=0/1、初期値は0。
<定数2>	等価回転/三軸直交/特異点通過=0/1/2、初期値は0。
<付随条件>	Wth節またはWthIf節を記述します。

【文例】

```
1 Mvr3 P1,P2,P3
2 Mvr3 P1,J2,P3
3 Mvr3 P1,P2,P3 Wth M_Out(17)=1
4 Mvr3 P3,(Plt 1,5),P4 WthIf M_In(20)=1,M_Out(21)=1
```

【解説】

(1) 円弧補間動作は、与えられた 3 点から円弧を求め、その円弧上を移動します。
(2) 姿勢は、始点から終点への補間となり、中心点の姿勢は影響しません。
(3) 現在位置と始点が一致しない場合は、自動的に始点まで直線補間 (三軸直交補間) します。
(4) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで関節補間で移動した後、残りの円弧補間を再開します。中断位置へ戻る補間方法 (関節補間 / 直線補間) は RETPATH パラメータで変更できます (442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください)。
(5) 等価回転 (定数 2=0) の場合、始点と終点の構造フラグが異なる場合は、実行時にエラーとなります。
(6) 定数 2 に三軸直交を指定した場合、定数 1 は無効でティーチング通りの姿勢に移動します。
(7) 定数 2 は姿勢補間のタイプを指定します。三軸直交とは、(X,Y,Z,J4,J5,J6) の座標系で補間を行う為、特異点近傍を通過する場合に効果的です。特異点通過は対応している機種が限定されています。詳細については 467 ページの「5.20 特異点通過機能について」を参照してください。
(8) 始点から終点までの中心角は、0 < 中心角 < 180 度です。
(9) 中心点から始点までの距離と、中心点から終点までの距離の差は、0.01mm 以内になるように位置を指定してください。
(10) 3 点が同一直線上にある場合や、始点と中心点、または終点と中心点が同一の場合はエラーとなります。
(11) 始点と終点が同一の場合や 3 点が同一の場合はエラーとならず、次の命令を実行します。
ただし、この時姿勢が異なる場合は姿勢のみの補間をおこないます。
(12) 常時実行プログラム内では使用できません。

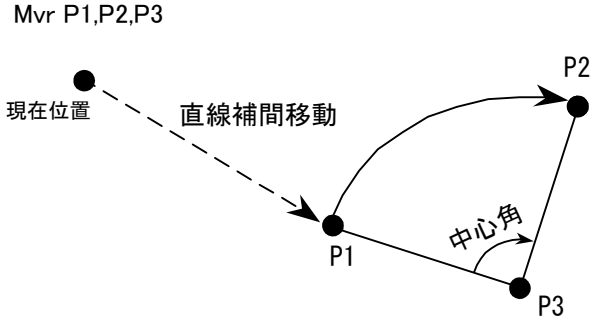


図 4-25 : 円弧補間の動作軌跡例その 3

◇◆◇垂直 5 軸ロボットにおいて直線(円弧)補間実行後の位置データ登録時の注意点◇◆◇

直線(円弧)補間において、動作開始位置の姿勢データと目的位置の姿勢データが、+180 度もしくは -180 度以上異なるとロボットは近回りで動作します。このとき垂直 5 軸ロボット(RV-4FJL)では、目的位置へ到達すると姿勢データは登録された値を表示していますが、実際には +360 度もしくは -360 度ずれた姿勢となっています。

例えば、A 軸の値が 0 度の位置から +200 度の位置へ直線補間させると、現在位置の姿勢データは +200 度と表示しますが、ロボットは -160 度の位置に動作しています。

この状態で位置データを登録しますと、実際とは +360 度もしくは -360 度異なった姿勢データが登録されます。プログラムのデバッグ操作(ステップ送り、MS ポジション移動)などで直線(円弧)補間動作後に位置データを補正する場合は、ご注意願います。

なお、この現象は直線(円弧)補間命令の引数 Type で遠回り動作を指定していただくか、位置データ登録時に一旦サーボオフしていただくことにより回避することが可能です。

Mvs（ムーブエス）

【機能】
現在位置から移動目的位置まで直線補間動作を行なう。

【書式 1】

Mvs □＜移動目的位置＞ [, ＜近接距離＞]
[□ Type □＜定数 1>, ＜定数 2>] □ [＜付随条件＞]

【書式 2】

Mvs □, ＜離脱距離＞[□ Type □＜定数 1>, ＜定数 2>] □ [＜付随条件＞]

【用語】

＜移動目的位置＞	補間動作の最終位置。位置型の変数と定数、または関節変数で記述します。
＜近接距離＞	この値を指定した場合、実際の移動目的位置は、ツール座標系のZ軸方向(+/-方向)に指定した距離だけ離れた位置へ動作します。
＜定数1＞	近回り/遠回り動作=0/1、初期値は0。
＜定数2＞	等価回転/三軸直交/特異点通過=0/1/2、初期値は0。
＜付随条件＞	Wth節またはWthIf節を記述します。
＜離脱距離＞	この値を指定した場合、現在の位置からツール座標系のZ軸方向(+/-方向)に指定した距離だけ離れた位置へ動作します。

【文例】

① 目的位置P1へ直線補間移動
```
1 Mvs P1
```

② 目的位置P1へ直線補間移動と同時に出力信号17をオン
```
1 Mvs P1,100.0 Wth M_Out(17)=1
```

③ 目的位置P4+P5(加算による相対演算位置)のツール座標系Z方向50mmへ直線補間移動中に、入力信号18がオンしたら、出力信号20をオンする
```
2 Mvs P4+P5, 50.0 WthIf M_In(18)=1, M_Out(20)=1
```

④ 現在地からツール座標系Z方向へ50mm直線補間移動
```
3 Mvs ,50
```

【解説】
(1) 直線補間動作は、ロボットの現在位置から移動目的位置までの制御点の軌跡が直線になるように移動する動作形態です。
(2) 姿勢は、始点から終点への補間となります。
(3) ＜近接距離＞や＜離脱距離＞で指定されるツール座標系は、ロボットの機種により Z の +/- 方向が異なります。詳しくは、432 ページの「5.6 標準ツール座標について」を参照ください。
「図 4-26：直線補間時の動作例」は、一例です。

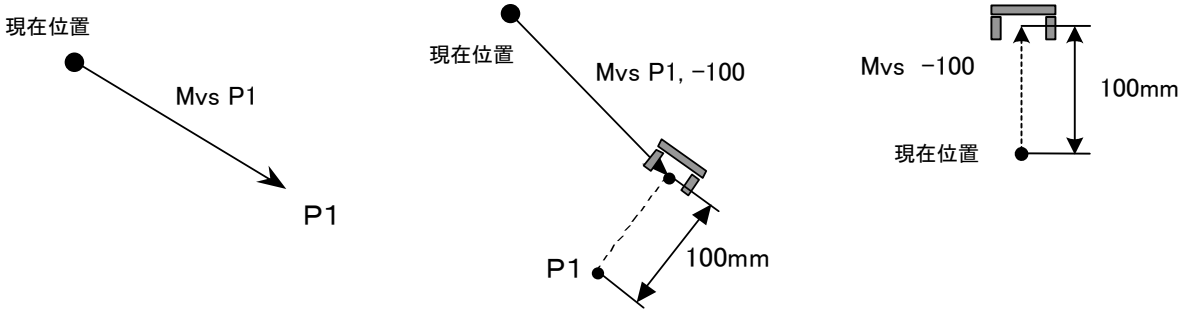
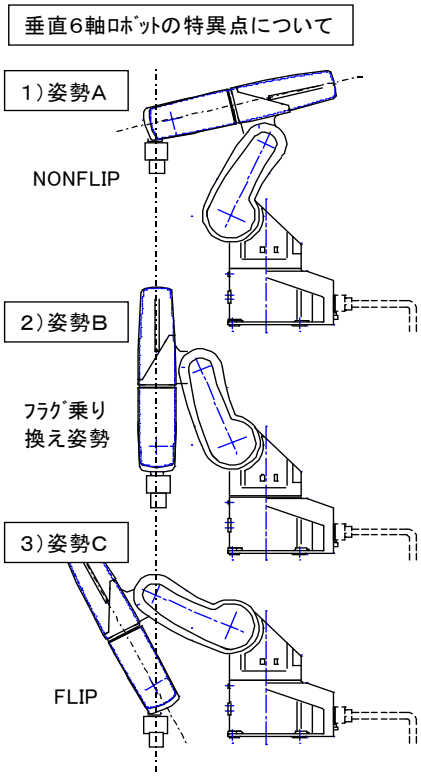


図 4-26：直線補間時の動作例

(4) 近回り動作とは、始点・終点間の姿勢において動作量が少ない方向に姿勢の補間を行ないます。
(5) 遠回り動作とは、始点・終点間の姿勢において動作量が多い方向に姿勢の補間を行ないます。
(6) 遠回り / 近回り指定した場合に、目的位置が動作範囲外となるときは、動作範囲外のエラーが発生します。
(7) 実行を中断して、ジョグで移動した後で再開した場合は、実行中断した位置まで関節補間で移動した後、目的位置に対して移動開始します。パラメータ「RETPATH」の設定によります。一部液晶搬送用ロボットでは初期値が異なります。442 ページの「5.10 中断時のジョグ送り後の自動復帰設定」を参照ください。
(8) 常時実行プログラム内では使用できません。
(9) 等価回転 (定数 2=0) の場合、始点と終点の構造フラグが異なる場合は、実行時にエラーとなります。
(10) 定数 2 に三軸直交を指定した場合、定数 1 は無効でティーチング通りの姿勢に移動します。
(11) 定数 2 は姿勢補間のタイプを指定します。三軸直交とは、(X,Y,Z,J4,J5,J6) の座標系で補間を行う為、特異点近傍を通過する場合に効果的です。特異点通過は対応している機種が限定されています。詳細については 467 ページの「5.20 特異点通過機能について」を参照してください。
(12) 特異点の解説



< 垂直 6 軸ロボットの場合 >
姿勢 A から姿勢 B を通って姿勢 C へは通常の直線補間 (Mvs) では動作できません。

姿勢 A、B、C とも J4 軸がゼロ度の場合に限りますが、姿勢 A は J5 軸 (手首軸) の構造ﾌﾗｸﾞが FLIP、姿勢 C は NONFLIP であるためです。また、姿勢 B は手首が伸びきっているため、同一直線上に J4 軸、J6 軸があるためロボットは直線補間の位置演算ができません。
このような姿勢で直線補間したい場合には Mvs のｺﾏﾝﾄﾞｵﾌﾟｼｮﾝで三軸直交 (Type 0,1) 方式 (または特異点通過 (Type 0,2) 方式) を使用します。
ただし、この 3 軸直交は姿勢 A と姿勢 C の J4,J5,J6 軸の関節角度を均等に補間するため厳密にいえば姿勢を保持しているわけではありません。よって移動する途中でロボットハンドの姿勢が前後に動くことが予想されます。
この場合は途中に 1 ポイント追加するとハンド姿勢の変化量を軽減すことができます。

またベース座標系の Z 軸上に J5 軸の中心点があり手首が真上を向いている場合も同様に特異点になります。これは J1 軸と J6 軸が同一軸上になりロボットの位置演算ができなくなるためです。

図 4-27 : 特異点その 1

◇◆◇垂直 5 軸ロボットにおいて直線 (円弧) 補間実行後の位置データ登録時の注意点◇◆◇
直線 (円弧) 補間において、動作開始位置の姿勢データと目的位置の姿勢データが、+180 度もしくは -180 度以上異なるとロボットは近回りで動作します。このとき垂直 5 軸ロボット (RV-F シリーズ) では、目的位置へ到達すると姿勢データは登録された値を表示していますが、実際には +360 度もしくは -360 度ずれた姿勢となっています。
例えば、A 軸の値が 0 度の位置から +200 度の位置へ直線補間させると、現在位置の姿勢データは +200 度と表示しますが、ロボットは -160 度の位置に動作しています。
この状態で位置データを登録しますと、実際とは +360 度もしくは -360 度異なった姿勢データが登録されます。プログラムのデバッグ操作 (ステップ送り、MS ポジション移動) などで直線 (円弧) 補間動作後に位置データを補正する場合は、ご注意願います。
なお、この現象は直線 (円弧) 補間命令の引数 Type で遠回り動作を指定していただくか、位置データ登録時に一旦サーボオフしていただくことにより回避することが可能です。

MvSpl（ムーブスプライン）

【機能】

指定したスプラインファイルに登録されている情報に基づき、スプライン補間を実行します。
（スプライン補間の詳細については 587 ページの「7.4 スプライン補間について」を参照してください。）
注）この命令はソフトウェアバージョン Ver.R5 以降（F-Q シリーズ）/Ver.S5 以降（F-D シリーズ）で使用できます。

【書式】

MvSpl □＜スプライン番号＞, ＜速度＞, ＜加減速距離＞
[, ＜フレーム変換＞[, ＜姿勢補間タイプ＞[, ＜フィルタ長＞]]]

【用語】

〈スプライン番号〉　動作させたい経路の情報を保持しているスプラインファイルの番号を、定数または数値変数で指定します。
設定範囲:1～99

〈速度〉　スプライン補間の動作速度を、定数または数値変数で指定します。
設定範囲:0.0より大きな値(mm/s)、詳細は【解説】(5)を参照してください。

〈加減速距離〉　スプライン補間の加減速距離を、定数または数値変数で指定します。
加速距離は、動作開始から指定した速度となるまでの加速に要する距離です。
減速距離は、指定した速度から終了位置に停止するまでの減速に要する距離です。
スプライン補間では、加速距離と減速距離は共通の設定となります。
設定範囲:0.0より大きな値(mm)

〈フレーム変換〉　フレーム変換の実行内容を、定数または数値変数で指定します。
設定範囲:
0:フレーム変換を実行しません。
1:スプラインファイルに設定された座標系を使ってフレーム変換を実行します。
2:SetCalFrm命令で設定された座標系を使ってフレーム変換を実行します。
省略時:フレーム変換を実行しません。

〈姿勢補間タイプ〉　姿勢の補間タイプを、定数または数値変数で指定します。
設定範囲:
0(等価回転動作)/1(三軸直交動作)
省略時:等価回転動作

〈フィルタ長〉　加減速動作にかかるフィルタ長を、定数または数値変数で指定します。
設定範囲:0.0～1000.0(ms)
省略時:100.0ms

【文例】

```
1 Ovrd 70              ' プログラムオーバーライドを 70% に設定
2 Mov P1               ' ポジション P1 の位置へ関節補間で移動
3 MvSpl 2, 50, 10      ' スプラインファイル 2 の経路点を通過するスプライン補間を実行
4 Mvs P2               ' ポジション P2 の位置へ直線補間で移動
5 MvSpl 2, 50, 10, 1   ' スプラインファイル 2 の経路点に対しフレーム変換を施した経路点を通過する
                       ' スプライン補間を実行
```

【解説】

(1)＜スプライン番号＞に対応したスプラインファイルに登録されている経路点データ等の情報に基づき、スプライン補間を実行します。経路点として登録されているロボットの位置・姿勢を通過する滑らかな曲線（スプライン曲線）を生成し、その曲線に沿って動作します。

(2) MvSpl 命令の実行開始時のロボットの現在位置とスプライン補間の開始位置がずれている場合は、開始位置まで直線補間で動作してからスプライン補間を開始します。

(3)＜スプライン番号＞に対応したスプラインファイルがコントローラに保存されていない場合は、エラー L2610（スプラインファイルを開くことができません）が発生します。

(4) スプライン補間の指令速度は、次式のとおり指定されます。
指令速度 ＝（MvSpl 命令の＜速度＞）×（Ovrd 命令）×（操作パネルのオーバーライド）
Spd 命令と JOvrd 命令の設定は使用されません。また、＜速度＞に状態変数 M_NSpd を設定しても最適速度制御モードは機能しません。
(5) ＜速度＞の設定範囲は、スプラインファイルに登録されている経路点間の距離に応じて変わります。詳細は、592 ページの「表 7-27：経路点に関するチェック」を参照してください。
(6) スプライン補間の加減速は＜加減速距離＞で指定します。Accel 命令の設定は使用されません。また、Oadl 命令で最適加減速制御を有効としていても、スプライン補間には適用されません。
(7) 加速動作は、スプライン補間の開始位置から＜加減速距離＞を移動すると＜速度＞に到達するように速度を生成します。減速動作は、＜速度＞から＜加減速距離＞を移動して終了位置で停止するように速度を生成します。
Ovrd 命令・操作パネルのオーバーライドが 100% より小さい値が設定されている場合は、＜加減速距離＞よりも短い移動量で指令速度に到達します。
（フィルタ等の影響により、実際には＜加減速距離＞で指定した値よりも長めの加減速距離となります。）

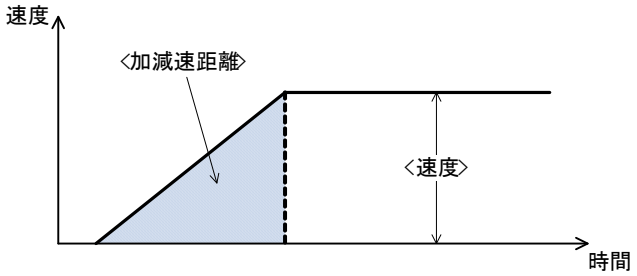


図 4-28：＜加減速距離＞による加速

(8) ＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」・「2(SetCalFrm 命令で設定した座標系を使用して実行)」を指定した場合は、指定された方法に基づいて逐次経路点データを変換し、その経路点を通るようにスプライン補間を実行します。
(9) ＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」を指定しているにも関わらず、スプラインファイルに座標系が設定されていない場合は、エラー L2042(フレーム変換座標が設定されていません)が発生します。
(10) ＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」を指定している場合に座標系が算出できない場合は、エラー L2041(フレーム変換座標が算出できません)が発生します。
(11) ＜フレーム変換＞に「2(SetCalFrm 命令で設定した座標系を使用して実行)」を指定すると、最後に実行した SetCalFrm 命令により設定された座標系を使ってフレーム変換を実行します。一度も SetCalFrm 命令が実行されておらず座標系が設定されていない場合は、エラー L2042(フレーム変換座標が設定されていません)が発生します。
(12) ＜姿勢補間タイプ＞の指定により、「0(等価回転動作)」と「1(三軸直交動作)」を切り替えることができます。
等価回転動作は、経路点間の姿勢変化が最小の角度となるように動作します。
三軸直交動作は、姿勢データを A,B,C 軸座標値ではなく J4,J5,J6 軸角度で生成します。そのため、経路点の姿勢データ(A,B,C 軸座標値)が同一であっても、動作中はロボットの姿勢が変化します。
(13) 構造フラグの乗り越え（特異点通過）には対応していません。経路点間で構造フラグが異なる場合は、エラー L2611（経路点の構造フラグが異なります）が発生します。
(14) ＜フィルタ長＞により加減速動作にかかるフィルタ特性を変更することができます。加速・減速時の動作をより滑らかにしたい場合は、フィルタ長を大きくします。ただし、大きくしすぎると動作が緩慢となり、スプライン補間が終了するまでの時間が長くなります。
(15) 経路点データに信号出力が設定されている場合は、対象の経路点を通過する際、指定された条件で汎用出力信号を出力します。
(16) ロボットの位置決め完了条件（Fine 命令、Fine J 命令、Fine P 命令）が指定されている場合、スプライン補間の終了位置で位置決め完了を確認します。
(17) Cnt 命令により連続動作が指定されていても、スプライン補間の開始位置・終了位置では連続動作は行いません。
(18) Def Act 命令（割り込み条件と処理の定義）の＜タイプ＞に「停止タイプ１」が指定されていても、スプライン補間実行中に割り込みが発生した場合は、「停止タイプ２」と同じ減速で停止します。
(19) 付加軸は、スプライン補間の開始位置から動作しません。途中の経路点に付加軸の位置が設定されていても、スプライン補間では使われません。

(20) MvSpl 命令に対して、ステップ送りを行うことはできますが、ステップ戻しを行うことはできません。エラー L2612（ステップ戻しは実行できません）が発生します。
(21) MvSpl 命令を実行するには、メカの制御権（GetM 命令）が必要です。
(22) 起動条件が ALWAYS・ERROR のスロットで MvSpl 命令を実行することはできません。エラー L2612（この起動条件では実行できません）が発生します。
(23) MvSpl 命令の引数に設定範囲外の値が設定されている場合は、エラー L3110（引数の値が範囲外です（MvSpl））が発生します。

⚠注意　＜速度＞に対してあまりにも短い＜加減速距離＞を指定しますと、急な加減速となり、ロボットの振動やサーボエラーの発生につながりますので、適切な加減速距離を設定してください。

【関連命令】
SetCalFrm（セット キャリブレーションフレーム）

【関連システム状態変数】
M_SplPno、M_SplVar

MvTune（ムーブチューン）

【機能】

用途に応じて最適な動作モードを指定することで、ロボットの動作性能を向上させることができます。
指定する動作モードは、下記 4 つの内いずれかを選択します。
なお、本命令では、LoadSet 命令で指定したハンド・ワーク条件に応じ、動作特性を最適化しています。
実際に使用するハンド･ワークの重量・形状・重心位置を正しく設定してください。

【書式】

```
MvTune □＜動作モード＞
```

【用語】

〈動作特性モード〉　ロボットの動作モード(1～4)を､定数または数値変数で指定します。
1:標準モード(初期値)
2:高速位置決めモード
3:軌跡優先モード
4:振動抑制モード

表 4-16：MvTune の動作モード

動作モード		特徴
1	標準モード （初期値）	メーカ標準設定です。 あらゆる用途で利用できる標準的な動作特性です。
2	高速位置決め モード	目的位置に到達するまでの時間を短縮します。 位置決め時間を短縮させ、作業効率を改善したい場合にご使用下さい。 （用途 ：トラッキング作業、パレタイズ作業　など）
3	軌跡優先 モード	補間動作の軌跡精度が向上します。 軌跡精度が重要とされる作業をおこなう場合にご使用下さい。 （用途 ：シーリング作業、溶着作業、バリ取作業　など）
4	振動抑制 モード	ロボットアームの振動（共振）を抑制する効果があります。 ワーク搬送中に振動が発生する場合にご使用下さい。 （用途 ：ウェハ搬送、精密部品搬送　など）

【文例】

```
LoadSet 1,1          ' ハンド 1・ワーク 1 に設定
MvTune 2             ' 動作モードを高速位置決めモードに変更
Mov P1               ' 高速位置決めモードで動作します
Mvs P1               ' 高速位置決めモードで動作します
MvTune 3             ' 動作モードを軌跡優先モードに変更
Mvs P3               ' 軌跡優先モードで動作します
```

【解説】

(1) LoadSet 命令で指定されたハンド・ワーク条件に応じて、最適な動作特性となるように調整します。実際に使用するハンド･ワークの重量・形状・重心位置を正しく設定してください。ハンド・ワーク条件が正しく設定されていない場合、十分な性能が出ない可能性があります。(456 ページの「5.16 ハンド、ワーク条件設定（最適加減速設定）について」を参照ください。)
(2) 電源 ON 直後は、標準モードが初期値として指定されます。
(3) プログラム終了(End 命令実行、または中断中のプログラムリセット）で、動作モードは標準モードに戻ります。ただし、CallP 命令で実行されたサブプログラムの End 命令、およびプログラム中断中の再始動（中断し、プログラム編集後の再起動も含む）では、その時の動作モードを維持します。

(4) 標準モードに対する各動作モードの違いを表 4-17 に示します。
表 4-17：MvTune 標準モードとの違い

動作モード	比較項目			
	目標位置到達時間	軌跡精度注 1)	振動抑制注 2)	負荷率注 3)
1：標準モード	○	○	○	○
2：高速位置決めモード	◎	○＋	○	△
3：軌跡優先モード	△	◎	○	○＋
4：振動抑制モード	○－	○－	◎	○

（注）表中の記号は性能比較レベルの度合いを表します。
◎：向上、○＋：やや向上、○：同一、○－：やや低下、△：低下

注 1）動作指令値に対するロボットの追従性能を比較
注 2）振動を誘発する外乱を抑制する能力について比較
注 3）モータの発熱量を比較

(5) 最適加減速制御（Oadl 命令、またはパラメータ ACCMODE で指定）が無効の場合は、本命令実行にて自動的に有効となります。また、本命令実行後に OadlOff を実行した場合、最適加減速制御のみ無効となり、動作モードは変更しません。
(6) 高速位置決めモードにおいて、加速・減速時の振動が標準モードに比べて大きくなる場合があります。支障がある場合は、標準モードを選択してください。
(7) 軌跡優先モードは中低速域の動作速度で最大の効果が得られるように調整されています。このため、高速で小さな円を描くような動作を行った場合、標準モードに比べ振動が増加する場合があります。この場合は Spd 命令を用い、振動が小さくなるように動作速度を下げてください。
(8) 振動抑制モードにおいて、ご使用条件によっては目的位置到達時のオーバーシュート量（行き過ぎ量）が大きくなる場合があります。位置決め完了を待つような動作（Fine 命令を用いた動作）では、タクトタイムが遅くなる可能性があります。（ワーク質量が大きい程、オーバーシュート量が大きくなります）
(9) ジョグ動作に対しては機能しません。

【関連命令】
LoadSet（ロードセット）、Oadl（オーエーディーエル）、Prec（プレック）

【関連パラメータ】
ACCMODE、HNDDAT0 ～ 8、WRKDAT0 ～ 8

Mxt（ムーブエクスターナル）

【機能】

コントローラ制御時間（約 7.1msec）毎に、イーサネットのネットワークで接続された外部のパソコンなどから絶対位置データを取り込んで直接移動します。

【書式】

Mxt <ファイル番号>, <指令位置データ型>[, <フィルタ時定数>]

【用語】

<ファイル番号>	OPENコマンドで割り当てた1～8の番号を記述します。 OPENコマンドで通信先を指定していないとエラーになり通信できません。 また､通信先以外からの受信データは無視されます。
<指令位置データ型>	パソコン等から指令する位置データの型を指定します。 直交/関節/ﾓｰﾀﾊﾟﾙｽの位置が指定できます。 0:直交座標データ 1:関節座標データ 2:モータパルス座標データ
<フィルタ時定数>	フィルタ時定数(msec)を指定します｡0を指定した時は､フィルタなしとなります｡(省略時は0) 受信位置データにフィルタをかけて鈍らせた指令値を作りサーボに出力します。

【文例】

```
10 Open "ENET: 192.168.0.2" AS #1     ' イーサネットの通信先の IP アドレスを設定する
20 Mov P1                             ' P1 へ移動
30 Mxt 1,1,50                         ' リアルタイム外部制御でフィルタ時定数を 50msec にして動作
40 Mov P1                             ' P1 へ移動
50 Hlt                                ' プログラムを中断
```

【解説】

(1) Mxt 命令を実行すると，ネットワーク接続されたパソコン等から動作制御のための位置指令を取り込むことができます。(1 対 1 通信）
(2) 動作制御時間（約 7.1msec）で 1 つ位置指令を取り込み、動作できます。
(3) Mxt 命令の動作
  1) コントローラで本命令が実行されると、コントローラは指令値受信可能状態となります。
  2) コントローラはパソコンから指令値を受信すると、受信した次の制御処理周期内で受信した指令値をサーボ出力します。
  3) 指令値をサーボに送った後、コントローラからパソコンに現在位置などのコントローラの情報を送信します。
  4) パソコンからコントローラに指令値を送信した場合のみ、コントローラからパソコンに返信します。
  5) データが来ない場合は、現在位置を保ちます。
  6) パソコンからのリアルタイム外部指令終了のコマンドを受信すると、Mxt 命令を終了します。
  7) オペレーティングパネルや外部入力から停止すると、Mxt 命令を中断し、再始動するまで送受信も中断します。
(4) タイムアウト指定は、パラメータ MXTTOUT でおこないます。
(5) 任意の指定（先頭ビット・ビット幅）した入出力信号を位置データと同時に 1 つ送受信できます。
(6) 動作制御時間で動作指令するためには、十分な処理速度を持ったパソコンが必要です。
(7) WindoswXP または 2000/Pentium III 1GHz 以上、メインメモリ 512K 以上、コンソールアプリケーションを推奨とします。

【関連命令】

Open（オープン）

Oadl（オーエーディーエル）

【機能】

ロボットハンドの負荷状態に応じて、最適な加減速度の設定（最適加減速制御）を自動的におこないます。
これによりロボットの動作時間（タクト）を短縮することが可能になります。
最適加減速時の加減速度は下記の計算式になります。
加減速度（秒）= 最適加減速度（秒）× Accel 命令 (%) × M_SetAdl(%)

※ 最適加減速度は Oadl 命令使用時に最適に算出される加減速時間です。

【書式】

Oadl □ <On/Off>

【用語】

<On/Off>	On :最適加減速度を開始
	Off:最適加減速度を終了

【文例】

```
1 Oadl On
2 Mov P1          ' 最大負荷で動作
3 LoadSet 1, 1    ' ハンド 1・ワーク 1 に設定
4 Mov P2          ' ハンド 1+ ワーク 1 の負荷で動作
5 HOpen 1         '
6 Mov P3          ' ハンド 1 の負荷で動作
7 HClose 1        '
8 Mov P4          ' ハンド 1+ ワーク 1 の負荷で動作
9 Oadl Off
```

※ パラメータ HNDHOLD1=0,1 の場合

【解説】

(1) LoadSet 命令で指定されたハンド条件・ワーク条件に応じて、最適な加減速で動作します。
(2) ハンド開閉時のワーク把持 / 未把持の設定はパラメータ HNDHOLD1 ～ 8 に設定されます。
(3) Oadl の初期設定はパラメータ ACCMODE によって変更可能です。(408 ページの「表 5-1：動作関連パラメーター覧表」を参照願います。)
(4) 一度 Oadl を On したら、再度 Oadl Off を実行するかプログラム End を実行するまで有効になります。
(5) ハンドやワークの状況によっては通常より動作時間が延びる場合もあります。
(6) LoadSet 命令、Oadl 命令と、HNDDAT1(0) ～ 8、WRKDAT1(0) ～ 8 パラメータの設定値により最適な加減速動作を行うことが可能です。これらのパラメータの工場出荷時の設定は最大負荷となっています。ご使用の条件に合わせてパラメータ値を変更ください。(456 ページの「5.16 ハンド、ワーク条件設定（最適加減速設定）について」を参照願います。)
(7) 軸単位の加減速度割合の値はパラメータ JADL によって決まっています。S シリーズは機種毎で値が異なっています。415 ページの「JADL」を参照してください。

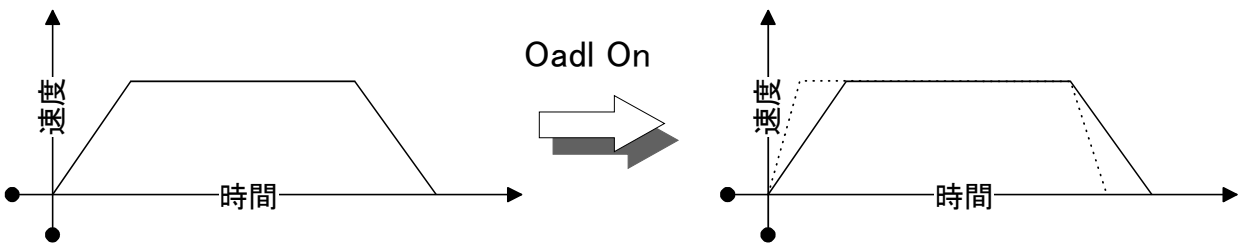


図 4-29：軽負荷時の加減速パターン

【関連命令】

Accel（アクセル）、LoadSet（ロードセット）、HOpen/HClose（ハンドオープン / ハンドクローズ）

【関連パラメータ】

HNDDAT0 ～ 8, WRKDAT0 ～ 8, HNDHOLD1 ～ 8, ACCMODE, JADL

On Com GoSub（オン コム ゴーサブ）

【機能】

指定された通信回線からの割り込みが発生した時、分岐する処理ルーチンの開始行を定義します。

【書式】

On □ Com[(< ファイル番号 >)] □ GoSub □ < 呼び出し先 >

【用語】

〈ファイル番号〉　通信回線に割り当てられた1～3の番号を記述します。
〈呼び出し先〉　ラベル名を記述します。

【文例】

ファイル番号 1 の通信回線 (COM1:) からの割り込みが発生したら、ラベル RECV の処理をおこなう。

```
1 Open "COM1:" As #1          ' 通信回線オープン
2 On Com(1) GoSub *RECV       ' 割り込みの定義
3 Com(1) On                   ' ファイル番号 1 の通信回線からの割り込み許可を設定
4 '
: ' << この間、通信の割り込みが発生した場合、ラベル *RECV へ分岐する。>>
11 '
12 Mov P1
13 Com(1) Stop                ' P1 から P2 へ移動中のみ、割り込みを禁止する。
14 Mov P2
15 Com(1) On                  ' P1 から P2 へ移動中に通信があった場合は、ここで割り込みが発生
16 '
: ' << この間、通信の割り込みが発生した場合、ラベル *RECV へ分岐する。>>
26 '
27 Com(1) Off                 ' ファイル番号 1 の通信回線からの割り込み禁止を設定
28 Close #1
29 End
  :
  :
40 *RECV                      ' 通信割り込み処理
41   Input #1, M0001          ' 受信した情報を M0001、P0001 に設定する。
42   Input #1, P0001
 :
50 Return 1                   ' 割り込みが発生した次の行へ制御を戻す
```

【解説】

(1) ファイル番号を省略した場合はファイル番号として 1 が採用されます。
(2) 割込みの優先順位はファイル番号の若い番号を優先します。
(3) ロボットが動作中に通信割り込みが発生すると、同一スロット内で動作しているロボットは停止します。Com Stop により割り込みを停止させ、ロボットを止めないようにすることもできます。
(4) 初期状態は割り込み禁止となっています。割り込みを有効にするには、本命令の後に Com On 命令を実行してください。
(5) サブルーチンではかならず Return で戻してください。GoTo で戻すと制御構造用メモリ（スタックメモリ）が減少してメモリ不足でエラーになります。

【関連命令】

Com On/Com Off/Com Stop（ｺﾑ ｵﾝ / ｵﾌ / ｽﾄｯﾌﾟ）、Return（リターン）、
Open（オープン）、Input（インプット）、Print（プリント）、Close（クローズ）

On･･･ GoSub（オン ゴーサブ）

【機能】
指定した値に対応するラベルのサブルーチンを呼び出す。

【書式】

On □＜式＞□ GoSub □ [＜呼び出し先＞][, [＜呼び出し先＞]]…

【用語】
〈式〉　何番目のラベルへ分岐するのかを数値演算式で指定します。
〈呼び出し先〉　ラベル名を記述します。 最大数は32です。

【文例】
入力信号 16 から 3 ビット分の値を M1 に設定し、その M1 の値（1 ～ 7）によって分岐する。
（M1 が 1 の時ラベル ABC1、2 の時ラベル Lsub、3、4、5 の時ラベル LM1_345、6、7 の時ラベル L67 を呼び出す）

```
1 M1 = M_Inb(16) And &H7
2 On M1 GoSub *ABC1,*Lsub,*LM1_345,*LM1_345,*LM1_345,*L67,*L67

100 *ABC1
101 ' M1=1 の時の処理を記述
102 Return                                  ' 必ず Return で戻します。

121 *Lsub
122 ' M1=2 の時の処理を記述
123 Return                                  ' 必ず Return で戻します。

170 *L67
171   ' M1=6 または M1=7 の時の処理を記述
172 Return                                  ' 必ず Return で戻します。

200 *LM1_345
201 ' M1=3、M1=4、M1=5 の時の処理を記述
202 Return                                  ' 必ず Return で戻します。
```

【解説】
(1)＜式＞の値で、どのラベルのサブルーチンを呼び出すかが決まります。
例えば、＜式＞の値が 2 の場合には、2 番めに記述してあるラベルを呼び出します。
(2)＜式＞の値が＜呼び出し先＞の個数より大きい場合は次行にジャンプします。例えば＜式＞の値が 5 で＜呼び出し先＞が 3 個しか書いていない場合は次行にジャンプします。
(3) 呼び出されたラベル名が使用されていない場合や、ラベル名が二重に定義されている場合は、実行時エラーとなります。
(4) サブルーチンではかならず Return で戻してください。GoTo で戻すとプログラム管理メモリ（スタックメモリ）が減少しメモリ不足でエラーになります。

＜式＞の値	処理＜制御＞
実数	値を四捨五入で整数に変換して分岐を行う
0、または行番号・ラベルの個数を超える場合	次の行に制御が移る
負の数または 32767 より大きい	実行時エラー

On…GoTo（オン ゴートゥー）

【機能】
指定した値に対応するラベルへ分岐します。

【書式】

On □＜式＞□ GoTo □ [＜分岐先＞][, [＜分岐先＞]]…

【用語】
〈式〉　何番目のラベルへ分岐するのかを数値演算式で指定します。
〈分岐先〉　ラベル名を記述します。 最大数は32です。

【文例】
数値変数 M1 の値（1 ～ 7）によって分岐する。
（M1 が 1 の時ラベル ABC1 へ、2 の時ラベル LJMP へ、3、4、5 の時はラベル LM1_345 へ、6、7 の時ラベル L67 へ分岐）

```
10 On M1 GoTo *ABC1,*LJMP,*LM1_345, *LM1_345, *LM1_345,*L67,*L67
11 ' M1 が 1 ～ 7 以外（0 や 8 ～）時、この行に制御が移る

100 *ABC1
101 ' M1=1 の時の処理を記述
102 '    :

110 ' M1=2 の時
111 *LJMP
112 '  M1=2 の時の処理を記述
113 '    :

170 *L67
171 '  M1=6 または M1=7 の時の処理を記述
172 '    :

200 *LM1_345
201 ' M1=3、M1=4、M1=5 の時の処理を記述
202 '    :
```

【解説】
(1) On GoSub の GoTo 版です。
(2) ＜式＞の値が＜分岐先＞の個数より大きい場合は次行にジャンプします。例えば＜式＞の値が 5 で＜分岐先＞が 3 個しか書いていない場合は次行にジャンプします。
(3) 呼び出された行番号 / ラベル名が使用されていない場合や、ラベル名が二重に定義されている場合は、実行時エラーとなります。

＜式＞の値	処理＜制御＞
実数	値を四捨五入で整数に変換して分岐を行う
0、または行番号・ラベルの個数を超える場合	次の行に制御が移る
負の数または 32767 より大きい	実行時エラー

Open（オープン）

【機能】
ファイル（通信回線含む）をオープンします。

【書式】

```
Open □ "<ファイル名>" □ [For　<モード>] □
        As □ [#]<ファイル番号>
```

【用語】
〈ファイル名〉　ファイル名を記述します。
・通信回線を使用する場合は　"〈通信回線名〉:"
・ファイルの場合"〈ファイル名〉"

ファイル種類	ファイル名	アクセス方法種類
ファイル	任意の 16 文字以内で記述	Input, Output, Append
通信回線	COM1: パラメータ「COMDEV」での設定 : COM8: パラメータ「COMDEV」での設定	省略 = ランダムモードのみ
	ENET: 192.168.0.2[注 1)]	Mxt

注 1）イーサネットインタフェースによってリアルタイム外部コントロールを使う場合の指定です。
"ENET :" の後に Mxt 命令によって絶対位置データを読み込む IP アドレスを指定します。

〈モード〉　ファイルへのアクセス方法を指定します。
・省略時:ランダムモード、通信回線を使用する場合は省略します。
・Input :入力モード既存ファイルから入力します。
・Output:出力モード(新規ファイル)、新規ファイルを作成し、それに出力します。
・Append:出力モード(既存ファイル)、既存ファイルの最後に追加出力します。
〈ファイル番号〉　1～8、の定数を記述します。

【文例】
① 通信回線

```
1 Open "COM1:" As #1      ' パラメータ：COMDEV の 1 番目の要素で設定された通信回線を、ファイル番号 1 としてオープンする
2 Mov P_01
3 Print #1,P_Curr         ' 現在位置を外部へ出力。"(100.00,200.00,300.00,400.00)(7,0)" の形式
4 Input #1,M1,M2,M3       ' "101.00,202.00,303.00" のアスキー形式で外部から受け取る。
5 P_01.X=M1
6 P_01.Y=M2
7 P_01.C=Rad(M3)          ' グローバルデータへコピーする。
8 Close                   ' オープンされた全てのファイルを閉じる
9 End
```

② ファイル操作　(R/C 内に "temp.txt" というファイルを作成し、"abc" と書込む)

```
1 Open "temp.txt" For Append As #1 ; temp.txt というファイルを Append モードでファイル番号 1 としてオープンします。
2 Print #1, "abc"                  ; ファイルに "abc" を書き込みます。
3 Close  #1                        ; ファイルを閉じます。
```

【解説】
(1)〈ファイル名〉に記述したファイルを指定したファイル番号にて開きます。
ファイルへの読み書きをおこなうにはこのファイル番号を使用します。
(2) 通信回線もファイルとします。

【関連命令】
Input（インプット）、Print（プリント）、Close（クローズ）

【関連パラメータ】
COMDEV

Ovrd（オーバーライド）

【機能】
ロボット動作のスピードを 1 ～ 100% にて指定します。プログラム全体にかかるオーバーライドです。

【書式】

Ovrd □＜オーバーライド＞

Ovrd □＜オーバーライド＞[, ＜上昇時オーバーライド＞[, ＜下降時オーバーライド＞]]

【用語】

〈オーバーライド〉	オーバーライドを実数で指定します。初期値は100です。 単位:[%](範囲:0.01～100.0) 数値演算式でも記述できます。0または100以上を設定するとエラーになります。
〈上昇時/下降時オーバーライド〉	アーチモーション命令(Mva)での上昇時､下降時のオーバーライド値を設定します。

【文例】

```
1 Ovrd 50
2 Mov P1
3 Mvs P2
4 Ovrd M_NOvrd          ' 初期値を設定
5 Mov P1
6 Ovrd 30,10,10         ' アーチモーション命令時の上昇下降時のオーバーライドを 10 に設定
7 Mva P3,3
```

【解説】
(1) Ovrd 命令は補間の種類に関係なく有効です。
(2) 実際のオーバーライドを以下に示します。
・関節補間動作時 =(操作パネル (T/B) のオーバーライド設定値)×(プログラムオーバーライド (Ovrd 命令))×(関節オーバーライド (JOvrd 命令))
・直線補間動作時 =(操作パネル (T/B) のオーバーライド設定値)×(プログラムオーバーライド (Ovrd 命令))×(直線指定速度 (Spd 命令))
(3) Ovrd 命令はプログラムオーバーライドのみを変化させます。100% がロボットの能力の最大となり、通常、システム初期値 (M_NOvrd) は 100% が設定されています。プログラム中、OvrD 命令が実行されるまでの指定オーバーライドはシステム初期値が採用されます。
(4) 一度 Ovrd 命令を実行すると、次に Ovrd 命令が実行されるか、プログラム End の実行、またはプログラムリセットするまでは指定されたオーバーライドが採用されます。End 文の実行、またはプログラムリセットで初期値に戻ります。

【関連命令】
JOvrd (ジェーオーバーライド)(関節補間用)、Spd (スピード)(直線補間、円弧補間用)

【関連システム状態変数】
M_NOvrd(システムの初期値)、M_Ovrd(現在の指定速度)、

Plt（パレット）

【機能】
パレット内の格子点位置を演算します。

【書式】

Plt □＜パレット番号＞, ＜格子点番号＞

【用語】
〈パレット番号〉　Def Plt分で設定済みの1～8の内のパレット番号を選びます。定数または変数で指定します。
〈格子点番号〉　求めたいパレット内の位置番号。変数または定数で指定します。

【文例】

```
1 Def Plt 1,P1,P2,P3,P4,4,3,1      '4 点パレットの定義 (P1,P2,P3,P4)
2 '
3 M1=1                             ' M1（カウンタ） 初期化
4 *LOOP
5   Mov PICK, 50                   ' ワーク取り出し位置上空 50mm へ移動
6   Ovrd 50
7   Mvs PICK
8   HClose 1                       ' ハンド閉
9   Dly 0.5                        ' ハンド閉完了 0.5 秒待ち
10   Ovrd 100
11   Mvs,50                        ' 現在位置上空 50mm へ移動
12   PLACE = Plt 1, M1             ' M1 番目の位置を計算
13   Mov PLACE, 50                 ' パレット上置き位置上空 50mm へ移動
14   Ovrd 50
15   Mvs PLACE
16   HOpen 1                       ' ハンド開
17   Dly 0.5
18   Ovrd 100
19   Mvs,50                        ' 現在位置上空 50mm へ移動
20   M1=M1+1                       ' カウンタ加算
21   If M1 <=12 Then *LOOP         ' カウンタが範囲内なら *LOOP から繰り返す
22   Mov PICK,50
23 End
```

【解説】
(1) Def Plt 文で定義したパレットの格子点の位置を演算します。
(2) パレット番号は、1 ～ 8 の最大 8 個が同時に定義可能です。
(3) 格子点の位置は、パレット定義の方向指定により異なるので注意してください。
(4) パレット定義文で定義された最大格子点番号をこえた格子点番号を指定すると、実行時エラーとなります。
(5) パレットの格子点を移動命令の目的位置に使用する場合は、下例のように括弧で括らないとエラーになります。

```
Mov (Plt 1, 5)
```

詳しくは 110 ページの「4.1.2 パレット演算」を参照してください。

【関連命令】
Def Plt（デフ パレット）

Prec（プレック）

【機能】
動作軌跡を向上させる場合に使用します。高精度モードの有効 / 無効を切り替えます。

【書式】

Prec □ <On/Off>

【用語】
<On/Off>　　On　:高精度モード有効時
　　Off:高精度モード無効時

【文例】
```
1 Prec On          ' 高精度モードを有効にする
2 Mvs P1
3 Mvs P2
4 Prec Off         ' 高精度モードを無効にする
5 Mov P1
```

【解説】
(1) 軌跡精度を高めて補間動作させたいとき、Prec On 命令で高精度モードを有効にします。
(2) Load Set 命令で指定されたハンド、ワーク条件に応じて、最適な特性となるように調整されています。ハンド、ワーク条件が正しく設定されていない場合、十分な性能が出ない可能性があります。(456 ページの「5.16 ハンド、ワーク条件設定（最適加減速設定）について」を参照ください。)
(3) Prec 命令は MvTune 命令と同一の処理となります。Prec Off は MvTune1(標準モード) に、Prec On は MvTune3(軌跡優先モード) に対応します。詳細については、MvTune 命令の頁を参照ください。

【関連命令】
LoadSet（ロードセット）、MvTune（ムーブチューン）

【関連パラメータ】
HNDDAT0 ～ 8、WRKDAT0 ～ 8

Print（プリント）

【機能】
ファイルにデータを出力します。データは全て ASCII 文字です。

【書式】

Print □ #< ファイル番号 > □ [, [< 式 >[;]]...[< 式 >[;]]]

【用語】
〈ファイル番号〉　Open命令で割り当てた1～8の番号を記述します。
〈式〉　数値演算式､位置演算式､文字列演算式を記述できます。

【文例】ファイル "temp.txt" へ出力する。

```
1 Open "temp.txt" For APPEND As #1 ' "temp.txt" をファイル番号 1 としてオープンする。
2 MDATA=150                        ' 数値変数 MDATA に 150 を代入します。
3 Print #1,"***Print TEST***"      ' 文字列 "***Print TEST***" を出力します。
4 Print #1                         ' 改行を出力します。
5 Print #1,"MDATA=",MDATA          ' 文字列 "MDATA="、続いて MDATA の値 (150) を出力します。
6 Print #1                         ' 改行を出力します。
7 Print #1,"****************"      ' 文字列 "****************" を出力します。
8 End
                                   ' プログラムを終了します。
```

出力結果は以下のようになります。

```
***Print TEST***

MDATA=150

****************
```

【解説】
(1)＜式＞が記述されていない場合は、改行コードのみを出力します。
(2) データの出力形式（例　参考）
＜式＞の値や文字列を出力する領域は、14 文字単位になっています。複数の値を出力する場合、＜式＞と＜式＞の区切り記号にカンマ(,)を使用します。出力時にはスペースが空きます。各領域単位の先頭に、セミコロン(;)を使用すると前に表示したものの直後に出力します。なお、出力データの最後には必ず改行コードがつきます。
(3) Open 文が実行されていない場合はエラーになります。
(4) ダブルクォート (") が含まれている場合、ダブルクォートまでしか出力されません。

例）
「1 M1=123.5
2 P1=(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0)」のとき
①「3 Print #1,"OUTPUT TEST", M1 , P1」と記述すると
OUTPUT TEST　123.5 (130.5,-117.2,55.1,16.2,0.0,0.0)(1,0) と、出力します。

②「3 Print #1,"OUTPUT TEST"; M1 ; P1」と記述すると
OUTPUT TEST 123.5(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0) と、出力します。

最後の＜式＞の直後にカンマまたはセミコロンを記述すると、改行を行わないで出力します。
③「3 Print #1,"OUTPUT TEST",
4 Print #1,M1;
5 Print #1,P1 」と記述すると
OUTPUT TEST 123.5(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0) と、出力します。

【関連命令】
Open（オープン）、Close（クローズ）、Input（インプット）

Priority（プライオリティ）

【機能】

マルチタスクでのプログラム運転は複数のプログラムの行を順番（初期値は 1 行づつ）に実行して動作しています。本命令はマルチタスクでプログラムを実行している場合の優先度（優先実行行数）を指定します。

【書式】

Priority □＜実行行数＞[, ＜スロット番号＞]

【用語】

＜実行行数＞	一度に実行する行数を定数または変数で指定します。1～31の数値を記述します。
＜スロット番号＞	1～32､省略時は現在のスロット番号になります｡定数または変数で指定します。

【文例】

スロット 1

```
10 Priority 3          ’現在のスロットの実行行数を 3 に指定します。
```

スロット 2

```
10 Priority 4          ’ このスロットの実行行数を 4 に設定します。
```

【解説】

(1) 指定した実行行数を実行するまで他のスロットのプログラムを実行しません。例えば上記文例のようにスロット 1 のプログラムにて Priority 3、スロット 2 のプログラムで Priority 4 と指定した場合は、まず、スロット 1 のプログラムを 3 行実行し、次にスロット 2 のプログラムを 4 行実行します。あとはこの繰り返しになります。
(2) 初期値は全スロットとも 1 です。よって、1 行実行するごとに次のスロットに移動します。
(3) 指定したタスクスロットにプログラムがない場合はエラーになります。
(4) 指定したタスクスロットのプログラムが運転中でも変更可能です。

RelM(リリースメカ)

【機能】
マルチタスク運転でタスクスロットにまたがってメカを制御する場合に使用します。GetM で取得したメカを開放する時に使用します。

【書式】

Relm

【文例】
① タスクスロット 1 からタスクスロット 2 を起動し、タスクスロット 2 の中でメカ 1 を制御する

タスクスロット 1

```
1 RelM                  ' スロット 2 でメカ 1 を制御するために、メカを開放する。
2 XRun 2,"10"           ' スロット 2 にプログラム 10 を選択
3 Wait M_Run(2)=1       ' スロット 2 の始動確認待ち
 :
```

タスクスロット 2(プログラム "10")

```
1 GetM 1                ' メカ 1 のリソースを取得する。
2 Servo On              ' メカ 1 のサーボオン
3 Mov P1
4 Mvs P2
5 Servo Off             ' メカ 1 のサーボオフ
6 RelM                  ' メカ 1 のリソースを開放
7 End
```

【解説】
(1) 現在取得しているメカリソースを開放します。
(2) メカを取得している状態で中断を行ってプログラムの実行を停止した場合、自動的に取得したメカリソースは開放されます。
(3) 常時実行プログラム内では使用できません。

【関連命令】
GetM(ゲットメカ)

Rem(レム)

【機能】
それ以降の文字列をコメントとします。

【書式】

Rem □[<コメント>]

【用語】
〈コメント〉　任意の文字列を記述します。
一行の範囲で記述が可能です。

【文例】

```
1 Rem ***MAIN PROGRAM***
2 ' ***MAIN PROGRAM***
3 Mov P1                        ' P1 に移動
```

【解説】
(1) Rem はシングルコーテーション(')で略記できます。
(2) 文例のステップ番号 3 のように命令の後に記述することもできます。

Reset Err(リセット エラー)

【機能】

コントローラに発生しているエラーをリセットします。初期状態では使用禁止となっています。警告エラー以外が発生している場合は通常のプログラムは運転できませんが、常時実行プログラムは運転可能です、この命令は常時実行プログラム内にて使用すると有用です。

【書式】

Reset Err

【文例】

常時実行プログラムにて実行する例

```
1 If M_Err=1 Then Reset Err' コントローラにエラーが発生した場合にエラーをリセットします。
```

【解説】

(1) パラメータ「SLT*」にて起動条件が常時実行(ALWAYS)設定にしてあるプログラムにてロボットのシステムエラーをリセットしたい場合に使用します。

(2) パラメータ「ALWENA」にて値を 0 から 1 に変更した後、コントローラの電源を再投入することで使用可能になります。

【関連パラメータ】

ALWENA

【関連システム状態変数】

M_Err(1：エラー発生中、0：エラーなし)

Return（リターン）

【機能】

①通常のサブルーチンからの Return の場合、GoSub された次のステップへ制御を戻します。
②割り込み処理用のサブルーチンからの Return の場合、割込みが発生したステップまたは次のステップへ制御を戻します。

【書式】

①通常のサブルーチンからの Return の場合。

```
Return
```

②割り込み処理用のサブルーチンからの Return の場合。

```
Return < 戻先指定番号 >
```

【用語】

〈戻先指定番号〉　割り込み発生時､割り込み処理実行後に制御を戻すステップを定数または変数で指定します。
0…割り込みが発生したステップへ制御を戻します。
1…割り込みが発生したステップの次のステップへ制御を戻します。

【文例】

① 通常のサブルーチンからの Return 例

```
1 ' ***MAIN PROGRAM***
2 GoSub *SUB_INIT       ' ラベル SUB_INIT へサブルーチンジャンプします。
3 Mov P1
 :
100 ' ***SUB INIT***    ' サブルーチン
101 *SUB_INIT
102   PSTART=P1
103   M100=123
104 Return              ' サブルーチンから呼ばれたステップの次のステップに戻ります。
```

② 割り込み処理用のサブルーチンからの Return 例
汎用入力信号 17 番の入力信号が ON 状態になったら、ラベル Lact のサブルーチンを呼び出す

```
1 Def Act 1,M_In(17)=1 GoSub *Lact      ' Act 1 の割り込みの定義をおこないます。
2 Act 1=1                               ' Act 1 を有効にします。
  :
10 *Lact                                ' Act 1 割り込み用サブルーチン
11   Act 1=0                            ' 割り込みを禁止します。
12   M_Timer(1)=0                       ' タイマーをゼロにセット
13   Mov P2                             ' P2 へ移動
14   Wait M_In(17)=0                    ' 入力信号 17 が Off するまで待ちます。
15   Act 1=1                            ' 割り込みを再設定します。
16 Return 0                             ' 割り込みが発生したステップへ戻ります。
```

【解説】

(1) GoSub で呼ばれた飛び先処理の最後に Return 命令を記述します。
(2) GoSub にて呼ばれていない場合に Return を実行した場合はエラーになります。
(3) GoSub で呼ばれた場合は必ず Return で戻します。GoTo 命令で戻すと制御構造用メモリ（スタックメモリ）が不足してエラーになります。
(4) 通常のサブルーチンからの Return に戻先指定番号があった場合、および割り込み処理用のサブルーチンからの Return に戻先指定番号がなかった場合は、実行時にエラーとなります。
(5) 割り込み処理から Return 1 で次のステップへ戻る場合は、割り込み処理内で割り込み禁止にしてください。割り込み禁止にせず、割り込み条件が成立したままの場合、再度割り込み処理を実行し次のス

テップへ戻るため、ステップが実行されずスキップすることがあります。割り込み処理に関しては 191 ページの「Def Act (デフ アクト)」を参照してください。

(6) 円、または円弧補間 (Mvc,Mvr,Mvr2,Mvr3) の実行中に割り込みが入り、Return 0 で元のステップに制御を戻した場合、ロボットは円、または円弧の始点に戻ってから再度円、円弧補間を実行します。

(7) アーチ補間の実行中に割り込みが入り、Return 0 で元のステップに制御を戻した場合、ロボットはその時の位置からアーチ補間をおこないます。

【関連命令】

Act (アクト)、GoSub(～ Return) (ゴーサブ リターン)、On… GoSub (オン ゴーサブ)、On Com GoSub (オン コム ゴーサブ)、Def Act (デフ アクト)

Select Case（セレクト ケース）

【機能】

条件式の値に従って複数のステートメントブロックのいずれかを実行します。

【書式】

```
Select □＜条件＞
            Case □＜式＞
                    [＜処理＞]
                    Break
            Case □＜式＞
                    [＜処理＞]
                    Break
                     :
            Default
                    [＜処理＞]
                    Break
End □ Select
```

【用語】

＜条件＞	数値演算式を記述します。
＜式＞	式を以下の形式で記述します。型は＜条件＞の型と同一であることが必要です。 ・Is＜比較演算＞＜定数＞ ・＜定数＞ ・＜定数＞To＜定数＞
＜処理＞	MELFA-BASIC Vで用意されている命令文(GoTo命令を除く)を記述します。

【文例】

```
1 Select MCNT
2   M1=10                   ' この行は実行されません。
3 Case Is <= 10             ' MCNT  <=10
4   Mov P1
5   Break
6 Case 11                   ' MCNT=11 OR MCNT=12
7 Case 12
8   Mov P2
9   Break
10 Case 13 To 18            '13  <=  MCNT  <=18
11   Mov P4
12   Break
13 Default                  ' 上記以外
14   M_Out(10)=1
15   Break
16 End Select
```

【解説】

(1) 条件が Case のいずれかと一致すると、Break または次の Case または Default または End Select までの処理が実行されます。条件が Case のいずれとも一致しなかった場合は、Default が記述されていればそのブロックを実行します。

(2) Default がない場合には、無処理で End Select の次行へジャンプします。

(3) Select Case と End Select 文は、必ず対応していなければなりません。Case ブロック内から GoTo 命令にて Select Case 外にジャンプすると制御構造用メモリ（スタックメモリ）が減少しますので連続で実行している場合はいつかエラーになります。

(4) Select Case に対応していない End Select 文を実行した場合は、実行時エラーとなります。

(5) Select Case の中に、さらに Select Case を記述することが可能です（ネスティングは 8 段まで可能）。

(6) Case 文の中に While ～ WEnd や For ～ Next を記述することは可能です。

(7) ＜式＞に比較演算（<、=、> 等）を用いる場合は、Case Is を使用します。

(8) Break は、省略することもできます。(Case 文の中の処理は(1)に従います)

Servo（サーボ）

【機能】
サーボモータ電源の入り切りを制御します。

【書式】
① 通常のプログラム

```
Servo □ <On/Off>
```

② 常時実行(ALWAYS)プログラム

```
Servo □ <On/Off>, < メカ番号 >
```

<On/Off>	On:サーボモータの電源を入れる場合 Off:サーボモータの電源を切る場合
<メカ番号>	常時実行プログラム内でのみ有効です。 1～3､定数または変数で記述します。

【文例】

```
1 Servo On                          ' サーボオン
2 *L20:If M_Svo<>1 GoTo *L20        ' サーボオン待ち
3 Spd M_NSpd
4 Mov P1
5 Servo Off
```

【解説】
(1) ロボット本体を全軸対象にサーボ電源の制御をおこないます。
(2) 付加軸がついている場合は付加軸のサーボ電源も対象になります。
(3) 常時実行プログラム内で使用する場合はパラメータ「ALWENA」にて値を 0 から 1 に変更しコントローラの電源を再投入することで有効になります。

【関連システム状態変数】
M_Svo(1 : ON、0 : OFF)

【関連パラメータ】
ALWENA

SetCalFrm（セット キャリブレーションフレーム）

【機能】

フレーム変換で使用する基準座標系を設定します。
本命令で２種類（フレーム変換前とフレーム変換後）の基準座標系を設定します。
注）この命令はソフトウェアバージョン Ver.R5 以降(F-Q シリーズ)/Ver.S5 以降(F-D シリーズ)で使用できます。

【書式】

SetCalFrm □＜位置 1>, ＜位置 2>, ＜位置 3>, ＜位置 4>, ＜位置 5>, ＜位置 6>

【用語】

<位置1>	フレーム変換前の基準座標系におけるX,Y,Z軸の原点となる位置データを､位置変数または位置定数で指定します。
<位置2>	フレーム変換前の基準座標系における+X軸上の点となる位置データを､位置変数または位置定数で指定します。
<位置3>	フレーム変換前の基準座標系のX-Y平面上における+Y方向の点となる位置データを､位置変数または数値定数で指定します。
<位置4>	フレーム変換後の基準座標系におけるX,Y,Z軸の原点となる位置データを､位置変数または位置定数で指定します。
<位置5>	フレーム変換後の基準座標系における+X軸上の点となる位置データを､位置変数または位置定数で指定します。
<位置6>	フレーム変換後の基準座標系のX-Y平面上における+Y方向の点となる位置データを､位置変数または数値定数で指定します。

【文例】

```
1 PR1=(0, 0, 0, 0, 0, 0)(0, 0)          ' フレーム変換前の基準座標系の原点位置
2 PR2=(1, 0, 0, 0, 0, 0)(0, 0)          ' フレーム変換前の基準座標系の +X 軸位置
3 PR3=(0, 1, 0, 0, 0, 0)(0, 0)          ' フレーム変換前の基準座標系の +Y 方向位置
4 PC1=(0, 0, 0, 0, 0, 0)(0, 0)          ' フレーム変換後の基準座標系の原点位置
5 PC2=(1, 1, 0, 0, 0, 0)(0, 0)          ' フレーム変換後の基準座標系の +X 軸位置
6 PC3=(-1, 1, 0, 0, 0, 0)(0, 0)         ' フレーム変換後の基準座標系の +Y 方向位置
7 SetCalFrm PR1, PR2, PR3, PC1, PC2, PC3 ' フレーム変換で使用する基準座標系を設定
                                         ' フレーム変換後の基準座標系は、変換前に対して
                                         ' Z 軸周りに +45 度回転した座標系
8 MvSpl 5, 50, 20, 2                    ' SetCalFrm 命令で設定した基準座標系を使って、スプライン
                                         ' ファイル 5 の経路点をフレーム変換し、その経路点を通過する
                                         ' スプライン補間を実行
```

【解説】

(1) フレーム変換で使用する２種類の基準座標系（フレーム変換前とフレーム変換後）を定義する位置データを設定します。１つの座標系を定義するために３つの位置データが必要となりますので、計６点の位置データを設定します。

下図の例では、位置データ PR1, PR2, PR3 でフレーム変換前の基準座標系 "Xfr-Zfr-Yfr"、位置データ PC1, PC2, PC3 でフレーム変換後の基準座標系 "Xfc-Zfc-Yfc" を定義します。

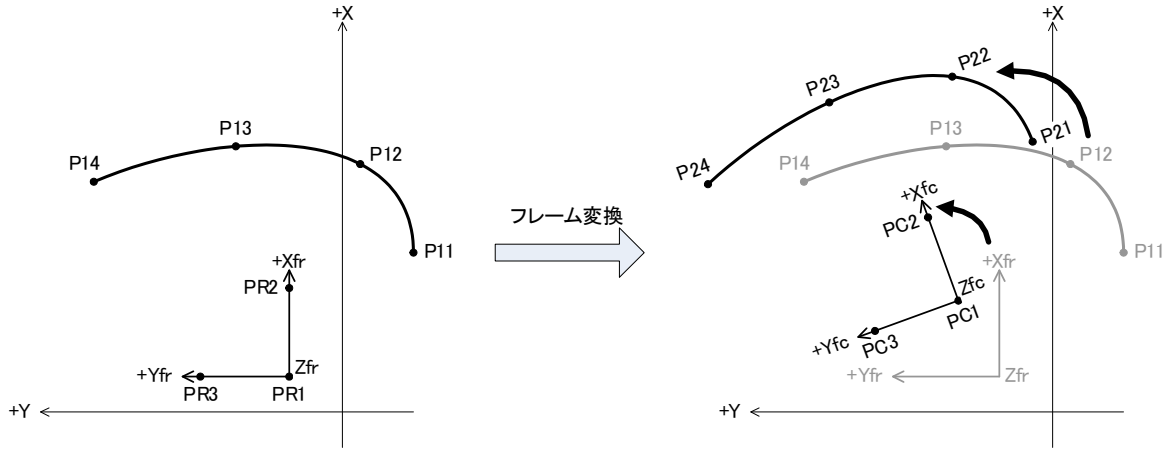


図 4-30：フレーム変換の例

(2) 座標系の定義には各位置データの X,Y,Z 軸座標値を使用します。A,B,C 軸座標値等、他の成分データは使用しません。
(3) 座標系を定義する３つの位置データの中に同一点が含まれている場合や、３つの位置データが直線上に並ぶ場合は、座標系を算出することができずエラー L2041（フレーム変換座標が算出できません）が発生します。
(4) コントローラの電源投入直後は、基準座標系は未設定の状態です。また、メインプログラムの End 命令・プログラムリセット操作により、基準座標系は未設定の状態に戻ります。
(5) SetCalFrm 命令を実行するには、メカの制御権（GetM 命令）が必要です。
(6) 起動条件が ALWAYS・ERROR のスロットで SetCalFrm 命令を実行することはできません。エラー L3287（起動条件が ERR、ALW の場合この命令は使用できません）が発生します

【関連命令】
MvSpl（ムーブスプライン）

【関連関数】
Fram

Skip（スキップ）

【機能】
プログラムの制御を次の行へ移します。

【書式】

Skip

【文例】

```
1 Mov P1 WthIf M_In(17)=1,Skip      ' 位置変数 P1 で示す位置へ関節補間で移動中に、入力信号
                                      M_IN(17) が ON になった場合、ロボットの補間動作を中断し、
                                      この命令の実行を打ち切り次のステップを実行します。
2 If M_SkipCq=1 Then Hlt'           スキップした場合はプログラムを中断します。
```

【解説】
(1) Wth 節または WthIf 節に使用します。この場合、そのステップの実行を中断し自動的に次のステップへ制御が移ります。Skip したかどうかは M_SkipCq で知ることができます。

【関連システム状態変数】
M_SkipCq(1：スキップした、0：スキップしていない)

Spd（スピード）

【機能】
ロボットの直線移動、円弧移動時の速度を指定します。また最適速度制御モードを指定します。

【書式】

```
Spd □ < 指定速度 >
Spd □ M_NSpd( 最適速度制御モード )
```

【用語】
〈指定速度〉　速度を実数で指定します。単位:[mm/s]

【文例】

```
1 Spd 100
2 Mvs P1
3 Spd M_NSpd          ' 初期値を設定 ( 最適速度制御モード )
4 Mov P2
5 Mov P3
6 Ovrd 80             ' 最適速度モード中の速度オーバーエラー対策
7 Mov P4
8 Ovrd 100
```

【解説】
(1) Spd 命令は直線補間、円弧補間時のみ有効です。
(2) 実際のオーバーライドは =(操作パネル (T/B) のオーバーライド設定値) × (プログラムオーバーライド (Ovrd 命令)) × (直線指定速度 (Spd 命令)) になります。
(3) Spd 命令は直線、円弧指定速度のみを変化させます。
(4) 指定速度に M_NSpd(初期値 : 値は 10000 です。) を指定した場合、ロボットは常に出しうる最高速度で動作しますので、線速は一定に保たれていません (最適速度制御)。
(5) 最適速度制御でもロボットの姿勢によってはエラーになる場合があります。もし、速度オーバーのエラーが発生した場合はその動作命令の前に Ovrd 命令を挿入し、その区間のみ速度をさげてご使用ください。
(6) プログラム中、Spd 命令が実行されるまでの指定速度はシステム初期値が採用されます。一度 Spd 命令を実行すると、次に Spd 命令が実行されるまでは指定された速度が採用されます。
(7) プログラム End 文を実行すると指定速度はシステム初期値に設定されます。

【関連システム状態変数】
M_NSpd(システムの初期値　最適速度制御モード 10000)
M_Spd(現在の指定速度)
M_RSpd(現在の実速度)

SpdOpt（スピードオプティマイズ）

【機能】

原点(X=Y=0、ロボットの特異点の１つ)の近くを通過する水平方向の直線補間動作で、速度オーバーとならないように速度を調整します。

注)本命令は以下のロボットでご利用できます。
機種：RH-3FHR シリーズ

【書式】

SpdOpt □ <On/Off>

【用語】

<On/Off>	On　:速度調整機能を有効にする Off:速度調整機能を無効にする

【文例】

```
1 Mov P1
2 SpdOpt On              ' 速度調整機能を有効
3 Mvs P2
4 Mvs P3
5 SpdOpt Off             ' 速度調整機能を無効
6 Mvs P6
```

【解説】

(1) 制御点の速度を保って直線補間で動作する場合、図 4-31 のように原点(ロボットの特異点の１つ)の近くを通過しようとすると、J1 軸は大きな速度で回転しなければならず、指定された速度によっては速度オーバーエラーが発生しますが、SpdOpt On を実行するとエラーが発生しないように自動的に速度を調整します。
例えば、指令速度 V で動作中、原点に近付き、そのままでは速度オーバーとなるような場合、図 4-32 の A のように自動的に速度を下げて速度オーバーの発生を防止します。その後、原点の近くを通り過ぎ速度を上げることができるようになると、図 4-32 の B のように指令速度 V に向かって加速を始めます。

(2) Ovrd または Spd 命令との関係
本機能での調整速度と比べてより遅い方で動作します。
・指定速度(Ovrd または Spd 命令)の方が遅い場合 …… 指定速度で動作
・指定速度(Ovrd または Spd 命令)の方が速い場合 …… 本機能での調整速度で動作

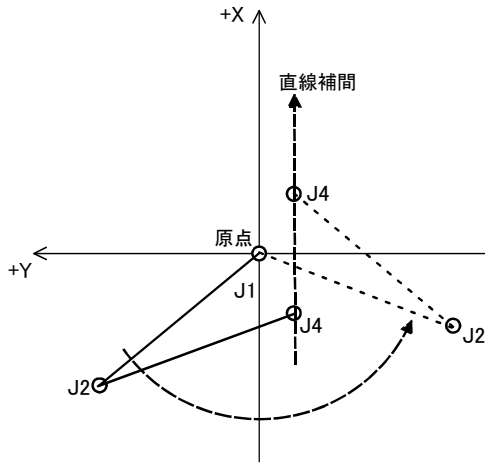


図 4-31：原点近での直線補間通過(例)

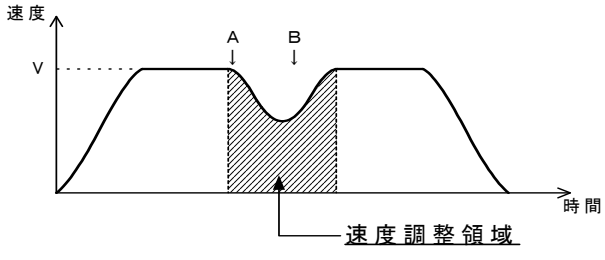
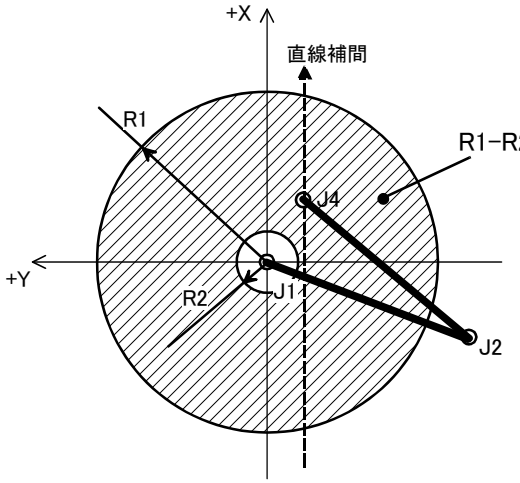


図 4-32：速度調整時の速度の様子

(3) 本命令が機能するのは直線補間動作のみです。関節補間や円弧補間では機能しません。
また、速度調整機能が有効となっていても、「図 4-33 速度調整領域と特異点領域」に示した領域を J4 軸が通過しない直線補間の場合は機能しません。



R1，R2 の距離
RH-3FHR シリーズ
R1=240.0mm，R2=1.0mm
R2 内：特異点領域（関節補間でのみ位置決め、及び通過可能）

図 4-33：速度調整領域と特異点領域

(4) 電源投入直後の速度調整機能の初期状態は、パラメータ SPDOPT で変更できます。このパラメータも使用可能機種を限定しています。
対象機種での初期値は SPDOPT=1（速度調整有効）となっています。
(5) End 命令、プログラムリセット操作を実行すると、速度調整機能の状態は電源投入直後の初期状態に戻ります。
(6) 速度調整機能が有効のとき、J4 軸が図 4-33 の特異点領域を通過するような直線補間を実行させようとすると、エラー 2804 が発生し、運転を中断します。
(7) 対象機種以外では、プログラムに本命令を記述しても無視されます。
(8) 原点の近くで Cnt 命令を有効にして軌跡接続をおこなったり、大きく姿勢変化するような直線補間動作を実行した場合、速度調整機能が有効であっても速度オーバーのエラーが発生することがあります。そのような場合は、軌跡接続をおこなう位置を原点から離したり、Ovrd 命令などで速度を調整してください。
(9) 水平方向にわずかに動作しながら垂直方向に大きく動作するような直線補間の場合、速度調整機能が有効になっていると無効の時に比べて動作速度が大きく低下する場合があります。そのような場合は、速度調整機能を無効とするか、関節補間（Mov 命令）で動作させてください。

【使用可能なロボット機種】

RH-3FHR シリーズ

【関連パラメータ】
SPDOPT

SplFWrt（スプラインフレームライト）

【機能】

指定したスプラインファイルにフレーム変換情報を登録します。

注）この命令はソフトウェアバージョン Ver.R6b 以降（F-Q シリーズ）/Ver.S6b 以降（F-D シリーズ）で使用できます。

【書式】

SplFWrt □ <スプライン番号>,<位置 1>,<位置 2>,<位置 3>,<位置 4>,<位置 5>,<位置 6>

【用語】

〈スプライン番号〉 フレーム変換情報を登録するスプラインファイルの番号を指定します。
設定範囲:1～99

〈位置1〉 フレーム変換で使用する基準座標系の原点を指定します。

〈位置2〉 フレーム変換で使用する基準座標系のX軸上の位置を指定します。

〈位置3〉 フレーム変換で使用する基準座標系のXY平面上の+Y方向の位置を指定します。

〈位置4〉 フレーム変換で使用する補正後の原点を指定します。

〈位置5〉 フレーム変換で使用する補正後のX軸上の位置を指定します。

〈位置6〉 フレーム変換で使用する補正後のXY平面上の+Y方向の位置を指定します。

【文例】

```
SplFWrt 1, PR1, PR2, PR3, PC1, PC2, PC3 'フレーム変換で使用する基準座標系（PR1, PR2, PR3）
                                         と補正後の座標系（PC1, PC2, PC3）をスプラインファ
                                         イル 1 に設定する。
```

【解説】

(1)<スプライン番号>で指定されたスプラインファイルにフレーム変換情報を登録します。

(2) フレーム変換で使用する 2 種類の基準座標系（フレーム変換前とフレーム変換後）を定義する位置データを設定します。1 つの座標系を定義するために 3 つの位置データが必要となります。

(3) 座標系の定義には各位置データの X,Y,Z 軸座標値を使用します。A,B,C 軸座標値等、他の成分データは使用しません。

(4) 座標系を定義する 3 つの位置データの中に同一点が含まれている場合や、3 つの位置データが直線状に並ぶ場合は、座標系を算出することができずエラー L2041（フレーム変換座標が算出できません）が発生します。

(5)<スプライン番号>に対応したスプラインファイルがコントローラに保存されていない場合は、エラー L2610（スプラインファイルを開くことができません）が発生します。

SplWrt（スプラインライト）

【機能】

指定したファイルの情報を含むスプラインファイルを作成します。

注）この命令はソフトウェアバージョン Ver.R6b 以降（F-Q シリーズ）/Ver.S6b 以降（F-D シリーズ）で使用できます。

【書式】

SplWrt □＜スプライン番号＞,＜ファイル名 1＞[,＜動作モード＞[,＜スプラインキャンセル角度＞[,＜ブロック長比率＞[,<Ex-T 座標番号＞[,＜ファイルバージョン＞]]]]]

【用語】

〈スプライン番号〉　作成するスプラインファイルの番号を指定します。
設定範囲:1～99

〈ファイル名〉　スプラインファイルに登録する経路点情報を含むファイル名を指定します。
任意の16文字以内で記述してください。

〈動作モード〉　スプライン補間での動作モード(線速一定/線速可変)を指定します。
設定範囲:
0:線速一定モード
1:線速可変モード
省略時:線速一定モードでスプラインファイルを作成します。

〈フレーム変換〉　フレーム変換の実行内容を､定数または数値変数で指定します。
設定範囲:
0:フレーム変換を実行しません。
1:スプラインファイルに設定された座標系を使ってフレーム変換を実行します。
2:SetCalFrm命令で設定された座標系を使ってフレーム変換を実行します。
省略時:フレーム変換を実行しません。

〈スプラインキャンセル角度〉スプライン補間をキャンセルする角度を指定します。
設定範囲:0～180(度)
省略時:120(度)

〈ブロック長比率〉　スプライン曲線を直線として動作させるブロック長比率を指定します。
設定範囲:0～100(倍)
省略時:8倍

〈Ex-T座標番号〉　制御点とするワーク座標の番号を指定します。
設定範囲:0～8
省略時:Ex-Tスプライン機能を無効とします。

〈ファイルバージョン〉作成するスプラインファイルのバージョンを指定します。
設定範囲:1～2
省略時:ファイルバージョンを2としてスプラインファイルを作成します。(コントローラが対応しているスプラインファイルの最新バージョンとします)

【文例】

```
SplWrt 1,"01.csv"          '"01.csv"に登録されている経路点情報に基づいたスプラインファイル1を
                           生成します。

SplWrt 2,"02.csv", 0, 120, 8, 1, 2"  '"01.csv"に登録されている経路点情報に基づいたスプライン
                                     ファイル1を生成します。
```

【解説】

(1)＜スプライン番号＞で指定されたスプラインファイルを作成します。
(2)＜ファイル名＞で指定されたファイルに登録されている経路点情報をスプラインファイルに登録します。

(3) <動作モード> で指定されたモードをスプラインファイルに登録します。省略時は 0(線速一定) としてスプラインファイルに登録されます。
(4) <スプラインキャンセル角度> で指定された角度をスプラインファイルに登録します。省略時は 120(度) としてスプラインファイルに登録されます。0 を指定した場合は、スプラインキャンセルは機能しません。
(5) <ブロック長比率> で指定された比率をスプラインファイルに登録します。省略時は 8(倍) としてスプラインファイルに登録されます。0 を指定した場合は、ブロック長比率は機能しません。
(6) <Ex-T 座標番号> に 1 ～ 8 を指定した場合は、WK1CORD ～ WK8CORD に設定された座標系データをスプラインファイルに登録します。0 を指定した場合は Ex-T スプライン機能を無効とします。省略時は 0(Ex-T スプライン機能無効) とします。
(7) <ファイルバージョン> で指定されたバージョンのスプラインファイルを作成します。省略時は 2(コントローラが対応しているスプラインファイルの最新バージョン) が登録されます。
(8) 本命令実行後、<ファイル名> で指定されたファイルはコントローラから削除されます。
(9) 別スロットのスプライン関連命令 (MvSpl、EMvSpl、SplPos、SplSpd、SplECord) で対象のスプラインファイルを開いている状態で本命令を実行した場合は、エラー L2610(スプラインファイルが作成できません) が発生します。
(10) <ファイル名> に指定された経路点ファイルを開くことができない場合は、エラー L2611(経路点ファイルを開くことができません) が発生します。
(11) <ファイル名> に登録された経路点情報がフォーマットと異なる (タグの数が異なる、タグ名が異なる) 場合は、エラー L2611(ファイルのフォーマットが異なります) が発生します。
(12) <ファイル名> に登録された経路点の数が異常 (4 点未満 /5001 点以上) の場合は、エラー L2611(経路点の個数不正) が発生します。
(13) <ファイル名> に登録された M_SplVar の値が範囲外の場合は、エラー L2615(M_SplVar の設定値が範囲外です) が発生します。
(14) <ファイル名> に登録されたトレランスの値が範囲外の場合は、エラー L2615(トレランスの設定値が範囲外です) が発生します。
(15) <ファイル名> に登録された出力信号の値が範囲外の場合は、エラー L2615(出力信号が範囲外です) が発生します。
(16) 引数の値が範囲外の場合は、L3110(引数の値が範囲外 (SplWrt)) が発生します。
(17) 引数の個数が異常 (多い / 少ない) の場合は、L4220(入力した命令文の構文に誤りがあります) エラーが発生します。
(18) コントローラにスプラインファイルを作成する容量がない場合は、エラー C7070(記憶容量不足です) が発生します。

【関連システム状態変数】
P_WkCord

【関連パラメータ】
WK1CORD ～ WK8CORD

Title（タイトル）

【機能】
プログラムにタイトルをつけます。別売のパソコンサポートソフトウェアにてコントローラの一覧表示欄に指定した文字を表示することが可能になります。

【書式】

Title □＜文字＞

【用語】
〈文字〉 タイトルとなるメッセージ。

【文例】
```
1 Title "ROBOT Loader Program"
2 Mvs P1
3 Mvs P2
```

【解説】
(1) プログラム上は一行に登録可能な文字数まで登録できますが、パソコンサポートソフトウェアのコントローラのプログラム一覧表示欄には 20 文字までの表示となります。

Tool（ツール）

【機能】

ツール変換データを指定します。ツール(ハンド)の長さ、メカニカルインタフェースからの制御点の位置、姿勢を設定します。

【書式】

Tool □＜ツールデータ＞

【用語】

〈ツールデータ〉　　位置演算式(位置定数、位置変数など)でツールデータを指定します。

【文例】

① 直接数値を設定

```
1 Tool (100,0,100,0,0,0)      ' ツール座標系で X 軸 100mm、Z 軸 100mm に制御点を変更
2 Mvs P1
3 Tool P_NTool                ' 制御点を初期値に戻す。（ﾒｶﾆｶﾙｲﾝﾀｰﾌｪｰｽ位置、ﾌﾗﾝｼﾞ面）
```

② 直交系の位置変数で設定

(PTL01 に (100,0,100,0,0,0,0,0) を設定すると①と同じ意味になります。)

```
1 Tool PTL01
2 Mvs P1
```

【解説】

(1) Tool 命令はダブルハンドを使用したシステムにて各ハンドごとの先端に制御点を設定したい場合などに使用します。ハンドが 1 種類の場合は Tool 命令ではなくパラメータ「MEXTL」にて設定してください。

(2) Tool 命令により変更したツール変換データは、パラメータ MEXTL に保存されて、コントローラの電源 OFF 後も保存されています。

(3) Tool 命令を実行するまではシステム初期値 (P_NTool) が採用されます。
一度 Tool 命令を実行すると、次に Tool 命令が実行されるまで、指定されたツール変換データが採用されます。メカ構造とは無関係に 6 軸三次元で演算します。

(4) ティーチング時と自動運転時にてツールデータが異なると思わぬ位置に動作する場合があります。必ず運転時とティーチングの設定を一致させるようにしてください。
また、ロボットの機種によって有効な軸成分が異なります。433 ページの「表 5-8：ロボットの形式に対するツール変換データの有効な軸成分」を参照して適切なデータを設定してください。

(5) M_Tool 変数を使用して MEXTL1 ～ 16 のパラメータをツールデータとして設定することも可能です。

【関連パラメータ】

MEXTL、MEXTL1 ～ 16　詳しくは432 ページの「5.6 標準ツール座標について」を参照。

【関連システム状態変数】

P_NTool(初期値)、P_Tool(現在の指定ツール変換データ)、M_Tool

Torq（トルク）

【機能】

各軸のトルクリミットを指定します。トルクリミットを指定することによって、ワーク等に過大な負荷（過重）をかけなくすることができます。トルク制限値率をオーバーすると誤差過大のエラーが出ます。

【書式】

Torq □＜軸番号＞, ＜トルク制限率＞

【用語】

〈軸番号〉　　軸番号を数値定数で指定します。(1から6) 定数または変数で指定します。
〈トルク制限率〉　軸から発生するカの制限を%で指定します。(1から100) 定数または変数で指定します。

【文例】

```
1 Def Act 1,M_Fbd>10 GoTo *SUB1,S  ' 指令位置とフィードバック位置の差が 10mm 以上で割込み
2 Act 1=1                          ' 割り込み有効
3 Torq 3,10                        ' トルク命令で 3 軸のトルクリミットを通常の 10% に設定
4 Mvs P1                           ' 動作
5 Mov P2
・
・
100 *SUB1
101 Mov P_Fbc                      ' 指令位置とフィードバック位置を一致させる
102 M_Out(10)=1                    ' 信号 10 番出力
103 End                            ' 差がでた時点で停止。
```

【解説】

(1) 指定した軸のトルクを制限し、指定以上のトルクがかからないように動作します。値は標準時のトルクに対する割合を指定します。標準のトルクはメーカにて決められた値になります。
(2) ロボットの機種によって、有効なトルク制限率は異なります。サーボモータ軸単位での設定になりますので、実際のロボット先端の制御点でのトルク制限率になるとは限りませんので、いろいろな比率でテストしてください。
(3) 本命令を実行して、トルクを制限した状態で停止させると指令位置とフィードバック位置がずれた状態で停止する場合（摩擦等により）があります。その場合動作再開時に誤差過大エラーが発生したりする可能性がありますので、上記例文のステップ番号 101 のように動作開始前にフィードバック位置へ移動させるようにプログラムしていただくと確実に動作します。
(4) 本命令は標準のロボットの軸のみ有効です。汎用サーボの軸（付加軸やユーザ定義のメカ）には使用できません。同様の動作をさせるには汎用サーボ側のパラメータを変更してください。

【関連システム状態変数】

M_Fbd、P_Fbc

Wait（ウェイト）

【機能】
変数が指定の値になるまで待機します。

【書式】

Wait □＜数値変数＞=＜数値定数＞

【用語】
〈数値変数〉　変数を指定します。入出力信号変数(M_In、M_Outなど)をよく用います。
〈数値定数〉　数値定数を指定します。

【文例】
① 信号の状態

```
1 Wait M_In(1)=1              ' 1 *L10:If M_In(1)=0 Then GoTo *L10 と同じ意味です。
2 Wait M_In(3)=0
```

② タスクスロットの状態

```
3 Wait M_Run(2)=1
```

③ 変数の状態

```
4 Wait M_01=100
```

【解説】
(1) 信号入力待ちや、マルチタスク実行状態でのインターロックに使用します。
(2) 指定の値になったら次の行へ移動します。
(3) マルチタスク実行状態で Wait 命令を複数のタスクで同時に実行するケースの場合、その間の処理時間(= タクトタイム)が伸び、システムに影響することがあります。このような場合は、Wait 命令を使用せずに If ～ Then 命令に置き換えてください。
例） 50 Wait M_ABC=0 → 50 *LBL50:If M_ABC<>0 Then GoTo *LBL50
(4) Wait 命令に記述する条件の個数は 1 つです。2 つ以上記述すると、誤った判定や実行時エラーとなります。
誤記例)Wait M_in(38)=1 Or M_IN(39)=1
⇒このような場合は、Wait 命令を使用せず、If ～ Then 命令に置き換えてください。
例)*Loop
If M_in(38)=1 Or M_IN(39)=1 Then *Next Else *Loop
*Next

While ～ WEnd(ホワイル エンド)

【機能】

While 文と WEnd 文の間のプログラムを、ループ条件を満たしている間繰り返し実行します。

【書式】

```
While □＜ループ条件＞
   :
WEnd
```

【用語】

〈ループ条件〉　　数値演算式を記述します。(162ページの「4.9 演算子」参照)

【文例】

①数値変数 M1 の値が -5 から +5 の範囲にある間、処理を繰り返し、範囲を越えた場合は、WEnd 文の次の行に制御が移ります。

```
1 While (M1>=-5) And (M1<=5)      ' 数値変数 M1 の値が -5 から +5 の間、処理を繰り返す。
2   M1=M1+1                       ' M1 に 1 を足します。
3   M_Out(8)=M1                   ' M1 の値を出力します。
4 WEnd                            ' While 文（ステップ 1）に戻ります。
5 End                             ' プログラムを終了します。
```

② While ～ WEnd の中から Break で WEnd の次行へ抜けることができます

```
1 While (M1>=-5) And (M1<=5)      ' 数値変数 M1 の値が -5 から +5 の間、処理を繰り返す。
2   M1=-(M1+1)                    ' M1 に 1 を足し、符号を反転します。
3   M_Out(8)=M1                   ' M1 の値を出力します。
4   If M_In(8)=1 Then Break       ' 入力信号 8 が ON していると、ステップ番号 6 へジャンプします。
5 WEnd                            ' While 文（ステップ 1）に戻ります。
6 If M_BrkCq=1 Then Hlt
```

【解説】

(1) While 文と WEnd 文の間のプログラムを繰り返し実行します。
(2) ＜式＞の結果が真である（0 でない）間は、While 文の次の行に制御が移り、処理が繰り返されます。
(3) ＜式＞の結果が真でない（0 である）場合は、WEnd 文の次の行に制御が移ります。
(4) While と WEnd との間から GoTo 命令にて強制的にジャンプすると制御構造用メモリ（スタックメモリ）が減少しますので連続で実行している場合はいつかエラーになります。While 文の条件成立にてループを抜けるようにプログラムしてください。
(5) While ～ WEnd の中に Break 文があれば、WEnd の次行へジャンプし、While ～ WEnd の処理から抜けます。

Wth（ウィズ）

【機能】
補間動作に処理を付加します。
【書式】

例）Mov P1 Wth □＜処理＞

【用語】
＜処理＞　付加される処理を記述します。記述できる命令は以下のものです。
①＜数値型ﾃﾞｰﾀB＞ ＜代入演算子＞ ＜数値型ﾃﾞｰﾀA＞[代入・信号修飾命令(162ページの「4.9 演算子」参照)]

【文例】
① P1 への移動開始と同時に、出力信号 17 番を数値変数 M1 が示す値 +2 秒間だけオンします。

```
10 Mov P1 Wth M_Out(17)=1 Dly M1+2     '
```

【解説】
(1) 移動命令の付随条件のみに使用できます。
(2) Wth 単独ではエラーとなります。
(3) 動作の開始と同時に処理を実行します。
(4) 他の割り込みとの優先順位関係は以下の通りです。
Com > Act > WthIf(Wth)

WthIf（ウィズイフ）

【機能】
補間動作命令に条件付で処理を付加します。
【書式】

例）Mov P1 WthIf □＜付加条件＞, ＜処理＞

【用語】
＜付加条件＞　処理を付加させる条件を記述します。(Act の条件式と同様)
＜処理＞　付加条件が成立した場合に付加される処理。(Wth と同様)
処理として記述できる命令を以下に示します。
①＜数値型ﾃﾞｰﾀ B＞ ＜代入演算子＞ ＜数値型ﾃﾞｰﾀ A＞
例） M_Out(1)=1、P1=P2
② Hlt 文
③ Skip 文

【文例】
① 入力信号 17 が ON したら停止する。

```
10 Mov P1 WthIf M_In(17)=1, Hlt
```

② 現在の指令速度が 200mm/sec を超えた場合、出力信号 17 を +2 秒間 ON する。

```
20 Mvs P2 WthIf M_RSpd>200, M_Out(17)=1 Dly M1+2
```

③ P3 へ直線補間移動中、到達率が 15% になったら、出力信号 1 を ON する。

```
30 Mvs P3 WthIf M_Ratio>15, M_Out(1)=1
```

【解説】
(1) 移動命令の条件付き付随条件のみに使用できます。
(2) 動作の開始と同時に条件監視を開始します。
(3) 処理のところに Dly 命令は記述できません。
(4) Hlt 命令と Skip 命令でロボット動作を停止させた場合、Def Act 命令の " 停止タイプ 1" または " 停止タイプ 2" と同じ減速停止となります (191 ページの「 Def Act (デフ アクト)」 参照)。Hlt はプログラム停止、Skip はプログラム運転を継続します。
パラメータ WTHFUNC を 0 に設定すると " 停止タイプ 1" を、1 にすると " 停止タイプ 2" を選択できます。初期設定は " 停止タイプ 1"(WTHFUNC=0) です。

XClr（エックスクリア）

【機能】

プログラム上から指定タスクスロットのプログラムの選択状態を解除します。指定タスクスロットに、新たなプログラムを実行できる状態にします。マルチタスク運転時に使用します。

【書式】

XClr □＜スロット番号＞

【用語】

〈スロット番号〉　1～32のスロット番号を指定します。定数または変数で指定します。

【文例】

```
1 XRun 2,"1"          ' ﾀｽｸｽﾛｯﾄ2にて1番のﾌﾟﾛｸﾞﾗﾑを実行

10 XStp 2             ' ﾀｽｸｽﾛｯﾄ2のﾌﾟﾛｸﾞﾗﾑを中断
11 Wait M_Wai(2)=1    ' ﾀｽｸｽﾛｯﾄ2のﾌﾟﾛｸﾞﾗﾑが中断するまで待つ
12 XRst 2             ' ﾀｽｸｽﾛｯﾄ2のﾌﾟﾛｸﾞﾗﾑの中断中状態を解除（ﾌﾟﾛｸﾞﾗﾑ選択可能状態に）する
13 Wait M_Psa(2)=1    ' ﾀｽｸｽﾛｯﾄ2がﾌﾟﾛｸﾞﾗﾑ選択可能になるまで待つ
14 XClr 2             ' ﾀｽｸｽﾛｯﾄ2のﾌﾟﾛｸﾞﾗﾑ選択状態を解除
15 End
```

【解説】

(1) 指定したスロットにプログラムが選択されていない場合はエラーになります
(2) 指定したスロットが運転中の場合はエラーになります。
(3) 指定したスロットが中断中の場合はエラーになります。
(4) 常時実行プログラム内で使用する場合はパラメータ「ALWENA」にて値を0から1に変更しコントローラの電源OFF → ONにすることで有効になります。

【関連命令】

XLoad（エックスロード）、XRst（エックスリセット）、XRun（エックスラン）、
XStp（エックスストップ）

【関連パラメータ】

ALWENA

XLoad（エックスロード）

【機能】

プログラム上から指定タスクスロットに指定プログラムをロードします。マルチタスク運転時に使用します。

【書式】

XLoad □＜スロット番号＞, ＜プログラム名＞

【用語】

＜スロット番号＞　1～32のスロット番号を指定します。定数または変数で指定します。
＜プログラム名＞　プログラム名を指定します。文字列定数で指定します(文字列変数は使用できません)。

【文例】

```
1 If M_Psa(2)=0 Then *LblRun            ' スロット 2 がプログラム選択可能状態かチェック
2 XLoad  2,"10"                         ' スロット 2 にプログラム 10 を選択
3 *L30:If C_Prg(2)<>"10" Then GoTo *L30 ' ロードされるまで待ちます。
4 XRun 2                                ' スロット 2 を始動
5 Wait M_Run(2)=1                       ' スロット 2 の始動確認待ち
6 *LblRun
7 ' スロット 2 が既に運転中の場合、ここより実行する
```

【解説】

(1) 指定したプログラムが存在しない場合は実行時エラーになります。
(2) 指定したプログラムが既に他のスロットに選択されている場合は実行時にエラーになります。
(3) 指定したプログラムが編集中の場合は、実行時にエラーになります。
(4) 指定したスロットが運転中の場合は、実行時エラーになります。
(5) プログラム名の指定はダブルクォーテーション(" プログラム名 ")で囲んで指定してください。
(6) 常時実行プログラム内で使用する場合はパラメータ「ALWENA」にて値を 0 から 1 に変更しコントローラの電源 OFF → ON にすることで有効になります。
(7) XLoad 実行直後に XRun するとプログラムロード中にエラーになる場合がありますので、必要な場合は文例のステップ番号 3 のようにロード完了チェックを行ってください。

【関連命令】

XClr（エックスクリア）、XRst（エックスリセット）、XRun（エックスラン）、
XStp（エックスストップ）

【関連パラメータ】

ALWENA

XRst（エックスリセット）

【機能】
プログラム上から指定したタスクスロットのプログラムが中断している場合、実行行を先頭行に戻します（プログラムリセット）。マルチタスク運転時に使用します。

【書式】

XRst □＜スロット番号＞

【用語】
〈スロット番号〉　1～32のスロット番号を指定します.定数または変数で指定します。

【文例】
```
1 XRun 2                ' 始動
2 Wait M_Run(2)=1       ' 始動完了待ち
   :
10 XStp 2               ' 停止
11 Wait M_Wai(2)=1      ' 停止完了待ち
  :
15 XRst  2              ' プログラムの実行開始行を先頭行にします。
16 Wait M_Psa(2)=1      ' プログラムリセット完了待ち
  :
20 XRun  2              ' 再開
21 Wait M_Run(2)=1      ' 再始動完了待ち
```

【解説】
(1) スロットが中断状態になっている場合にのみ有効です。
（スロットが運転中の場合や、プログラム未選択時はエラーになります。）
(2) 常時実行プログラム内で使用する場合はパラメータ「ALWENA」にて値を 0 から 1 に変更しコントローラの電源 OFF → ON にすることで有効になります。

【関連命令】
XClr（エックスクリア）、XLoad（エックスロード）、XRun（エックスラン）、
XStp（エックスストップ）

【関連パラメータ】
ALWENA

【関連システム状態変数】
M_Psa(スロット番号)(1：プログラム選択可能、0：不可)
M_Run(スロット番号)(1：実行中、0：実行中以外)
M_Wai(スロット番号)(1：中断中、0：中断中以外)

XRun（エックスラン）

【機能】

プログラム上から指定したプログラムを並列実行させます。マルチタスク運転時に使用します。

【書式】

XRun □＜スロット番号＞[, ["＜プログラム名＞"][, ＜運転モード＞]]

【用語】

＜スロット番号＞	1～32のスロット番号を指定します。定数または変数で指定します。
＜プログラム名＞	プログラム名を指定します。文字列定数または文字列変数で指定します。 スロットに既にプログラムが設定されている場合は省略可能です。
＜運転モード＞	0=連続運転、 1=サイクル停止運転、省略時は現状の運転モードになります。定数または変数で指定します。

【文例】

① XRun 命令で動作プログラムを指定した場合（連続運転）

```
1 XRun  2,"1"          ' プログラム 1 をスロット 2 で始動
2 Wait M_Run(2)=1      ' 始動完了待ち
```

② XRun 命令で動作プログラムを指定した場合（サイクル運転）

```
1 XRun  3,"2",1        ' プログラム 2 をスロット 3 でサイクル運転モードで始動
2 Wait M_Run(3)=1      ' 始動完了待ち
```

③ XLoad 命令を使って動作プログラムを指定した場合（連続運転）

```
1 XLoad 2, "1"                              ' プログラム 1 をスロット 2 で選択
2 *LBL: If C_Prg(2)<>"1" Then GoTo *LBL     ' ロード完了まで待つ
3 XRun 2                                    ' スロット 2 を始動
```

④ XLoad 命令を使って動作プログラムを指定した場合（サイクル運転）

```
1 XLoad 3, "2"                              ' プログラム 2 をスロット 3 で選択
2 *LBL:If C_Prg (3)<>"2" Then GoTo *LBL     ' ロード完了まで待つ
3 XRun 3,,1                                 ' スロット 2 をサイクル運転モードで始動
```

【解説】

(1) 指定したプログラムが存在しない場合は実行時エラーになります。
(2) 指定したスロット番号が既に使用されている場合は実行時にエラーになります。
(3) タスクスロットにプログラムがロードされていない場合は本命令にてロードをおこないますので、XLoad 命令を実行しなくても運転が可能です。
(4) プログラムの途中で停止した「中断中」状態で XRun を行なうと継続実行を行ないます。
(5) プログラム名の指定はダブルクォーテーション（" プログラム名 "）で囲んで指定してください。
(6) 運転モードは省略時は現状の運転モードで実行されます。
(7) 常時実行プログラム内で使用する場合はパラメータ「ALWENA」にて値を 0 から 1 に変更しコントローラの電源を再投入してください。
(8) XLoad 実行直後に XRun するとプログラムロード中のでエラーになる場合がありますので、必要な場合は文例③のステップ番号 2、④のステップ番号 2 のようにロード完了チェックを行ってください。

【関連命令】

XClr（エックスクリア）、XLoad（エックスロード）、XRst（エックスリセット）、
XStp（エックスストップ）

【関連パラメータ】

ALWENA

【関連システム状態変数】

M_Run(スロット番号)(1：実行中、0：実行中以外)

XStp（エックスストップ）

【機能】
プログラム上から指定したタスクスロットのプログラム運転を中断します。もし指定したタスクスロットでロボットを動作させている場合はロボットは停止します。マルチタスク運転時に使用します。

【書式】

XStp □＜スロット番号＞

【用語】
〈スロット番号〉　1～32のスロット番号を指定します.定数または変数で指定します。

【文例】
```
1  XRun  2              ' 実行
  :
10 XStp  2              ' 停止
11 Wait M_Wai(2)=1      ' 停止完了待ち
  :
20 XRun  2              ' 再開
```

【解説】
(1) 既に中断状態になっている場合には、エラーは発生しません。
(2) XStp は常時実行プログラムも停止させることが出来ます。
(3) 常時実行プログラム内で使用する場合はパラメータ「ALWENA」にて値を 0 から 1 に変更しコントローラの電源を再投入することで有効になります。

【関連命令】
XClr（エックスクリア）、XLoad（エックスロード）、XRst（エックスリセット）、
XRun（エックスラン）

【関連パラメータ】
ALWENA

【関連システム状態変数】
M_Wai(スロット番号)(1：中断中、0：中断中以外)

代入

【機能】

変数または配列変数に演算の結果を代入します。

【書式】

< 変数名 >=< 式 1>

パルス代入の場合

< 変数名 >=< 式 1> Dly < 式 2>

【用語】

<変数名>	値を代入したい変数名を記述します。 （変数の種類は142ページの「4.4.15 変数」参照）
<式1>	代入値。数値演算式を記述します。
<式2>	パルスタイマー。数値演算式を記述します。

【文例】

① 変数演算結果の代入

```
10 P100=P1+P2*2
```

② 信号の出力

```
20 M_Out(10)=1          ' 出力信号 10 を ON する
```

③ 信号のパルス出力

```
30 M_Out(17)=1 Dly 2.0  ' 出力信号 17 を 2 秒間 ON する
```

【解説】

(1) パルス出力用は次の行以降の命令の実行と並列に実行されます。
(2) パルス出力は、指定時間経過後に反転出力します。M_Outb,M_Outw にてパルス出力した場合は 8 ビット単位、16 ビット単位で反転しますのでご注意ください。任意ビット幅での反転はできません。
(3) パルス出力は、指定時間中に End 命令やプログラムの最終行が実行された場合には、指定時間経過を待たずにプログラムの実行は終了します。ただし、指定時間後に出力は反転します。

（ラベル）

【機能】

分岐先であることを表します。

【書式】

*< ラベル名 >

*< ラベル名 >[: < 命令行 >]

【用語】

〈ラベル名〉　英字から始まる文字列を記述します先頭は必ず英文字としてください。最高16文字(*を含んで最高17文字)以内で記述してください。

〈命令行〉　ラベルの後のコロン(:)に続けて命令行を記述できます。

【文例】

```
1 *SUB1
2 If M1=1 Then GoTo *SUB1
3 *LBL1:If M_In(19)=0 Then GoTo *LBL1      '入力信号 10 番が ON するまでステップ番号 3 で待ちます。
```

【解説】

(1) プログラムの中で参照していなくてもエラーにはなりません。

(2) 一つのプログラム内に、同一ラベルを複数定義すると二重定義エラーになります。

(3) 予約語をラベルに使うことはできません。

(4) ラベル名の 2 文字目にアンダースコア(″_″)がついた名称は、システム外部変数の形式のため使用できません。例えば「 *A_ 」「 *B_ 」「 *Z_ 」などは文法エラーとなります。
例外として、「 *L_ 」は、文法エラーにはなりませんが、実行時にエラーとなります(システム予約のため)。
なお、ラベル名の 3 文字目以降のアンダースコアは英数字と同等の扱いとなり、自由に使うことができます。

(5) ラベルの後のコロン(:)に続けて命令行を記述できます。ただし、命令行の後にコロンを記述して再度命令行を記述することはできません。

4.14 ロボット(システム)状態変数の詳細解説

4.14.1 記載項目の見方

【機能】　　：変数の機能を示します。
【書式】　　：変数の引数の入力方法を示します。[ ]は省略可能を意味します。
　　システム状態変数（ロボット状態変数）は参照、代入以外に条件式でも使用可能ですが、書式の例では単純な表記をするために参照または代入のみ記述にしてあります。
【文例】　　：変数を使ったプログラム例を示します。
【用語】　　：引数の意味、範囲などを示します。
【解説】　　：詳細な機能や、注意事項を示します。
【関連命令】　：関連する命令語を示します。
【関連パラメータ】　：関連するパラメータを示します。
【関連システム状態変数】：関連するシステム変数を示します。

4.14.2 各ロボット状態変数(システム状態変数)の説明

以下に、各変数の説明をアルファベット順に示します。

C_Com

【機能】

Open（オープン）命令でオープンする回線のパラメータを設定します。通信先を変更する場合に使用します。

【書式】

例）C_Com（<通信回線番号>）= "ETH:<サーバー側 IP アドレス>[,<ポート番号>]"

【用語】

ETH：対象がイーサネットであることを示す識別子

<通信回線番号>	Open 命令で指定される COM の番号（回線種類はパラメータ COMDEV で割り当てます）1～8 を指定します。
<サーバー側 IP アドレス>	サーバー側の IP アドレス（省略不可）
<ポート番号>	サーバー側のポート番号（省略時は、パラメータ NETPORT の設定値を使用）

【文例】

パラメータ COMDEV の第 2 要素に "OPT12" を設定した時の例

```
1 C_Com(2)="ETH:192.168.0.10,10010"    ' 通信回線 COM2 に対応する通信先の IP アドレスを
                                         192.168.0.10 に、ポート番号を 10010 に設定
2 Open "COM2:" AS #1                   ' 上記回線をオープン
3 If M_Open(1)<>1 Then 110             ' サーバーに接続できないときは、ループする
4 Print #1, "HELLO"                    ' 文字列を送信
5 Input #1, C1$                        ' 文字列を受信
6 Close #1                             ' 回線クローズ
7 C_Com(2)="ETH:192.168.0.11,10011"    ' 通信回線 COM2 に対応する通信先の IP アドレスを
                                         192.168.0.11 に、ポート番号を 10011 に設定
8 Open "COM2:" AS #1                   ' 上記回線をオープン
9 If M_Open(1)<>1 Then 170             ' サーバーに接続できないときは、ループする
10 Print #1, C1$                       ' 文字列を送信
11 Input #1, C2$                       ' 文字列を受信
12 Close #1                            ' 回線クローズ
13 Hlt                                 ' プログラム一時停止
14 End                                 ' 終了
```

【解説】

(1) パラメータ NETHSTIP、NETPORT でコントローラの通信相手先を指定して、その通信相手先を変更することがなければ、本コマンドを使う必要はありません。
(2) イーサネットのデータリンクのクライアント時のみ有効な機能です。
(3) Open（オープン）命令の通信パラメータを設定しますので、Open 命令より前に実行する必要があります。
(4) 電源投入時は、パラメータ NETHSTIP、NETPORT で指定された設定値が使用されます。本命令を実行すると、一時的に同パラメータで指定された値を変更します。電源を切るまで有効です。電源を再投入すると、パラメータで設定した値に戻ります。
(5) Open 命令の後に実行しても、そのときオープンしている状態は変化しません。そのような時は一度 Close（クローズ）命令で回線を閉じてから、再度 Open 命令を実行する必要があります。
(6) 書式が間違っている場合は、プログラム編集時ではなく、プログラム実行時にエラーとなります。

【関連パラメータ】

NETHSTIP、NETPORT

C_Date

【機能】
現在の日付を、年 / 月 / 日の形式で返します。

【書式】

例) < 文字列変数 >=C_Date

【文例】
```
1 C1$=C_Date              'C1$ には"2007/04/01" が入ります。
```

【解説】
(1) 現在の日付が入ります。
(2) 読み込み専用です。日付を設定する場合は T/B にて設定してください。

【関連システム状態変数】
C_Time

C_Maker

【機能】
コントローラのメーカ側の情報を返します。

【書式】

例) < 文字列変数 >=C_Maker

【文例】
```
1 C1$=C_Maker             'C1$ には"COPYRIGHT2007....." が入ります。
```

【解説】
(1) コントローラのメーカ側の情報を返します。
(2) 読み込み専用です。

【関連システム状態変数】
C_Mecha

C_Mecha

【機能】
制御権を獲得しているメカ名（ロボットの形名）を返します。

【書式】

例）<文字列変数>=C_Mecha[(<数値>)]

【用語】
<文字列変数>　代入する文字列変数を指定します。
<数値>　1～32、タスクスロット番号を入れます。省略時は現在のスロットです。

【文例】
```
1 C1$=C_Mecha(1)        'C1$ には"RH-3FH5515-D"が入ります。
                        （スロット1がメカ1("RH-3FH5515-D")の制御権を獲得している場合）
```

【解説】
(1) 指定したタスクスロットが制御権を獲得しているメカ名（ロボットの形名）を返します。
(2) メカの制御権を獲得していないタスクスロットを指定した場合は、""（なし）を返します。
(3) 読み込み専用です。

【関連命令】
GetM（ゲットメカ）、RelM（リリースメカ）

C_Prg

【機能】
選択されているプログラム名（番号）を返します。

【書式】

例）<文字列変数>=C_Prg[(<数値>)]

【用語】
<文字列変数>　代入する文字列変数を指定します。
<数値>　1～32、タスクスロット番号を入れます。省略時は現在のスロットです。

【文例】
```
1 C1$=C_Prg(1)          'C1$ には"10"が入ります。（プログラム名が"10"の場合）
```

【解説】
(1) 指定したタスクスロットに設定（ロード）されているプログラム名（番号）が入ります。
(2) シングルタスクで使用している場合のタスクスロット番号は1になります。
(3) 操作パネルにて設定した場合は、そのプログラム名（番号）になります。
(4) 読み込み専用です。
(5) プログラムがロードされていないタスクスロットを指定した場合は実行時にエラーとなります。

C_Time

【機能】
現在の時間を、時 / 分 / 秒の形式で返します (24 時間表記)。

【書式】

例) < 文字列変数 >=C_Time

【文例】
```
1 C1$=C_Time              'C1$ には "01/05/20" が入ります。
```

【解説】
(1) 現在の時刻を返します。
(2) 読み込み専用です。
(3) 時間の変更は T/B にておこなってください。

【関連システム状態変数】
C_Date

C_User

【機能】
パラメータ "USERMSG" に設定されているデータを返します。

【書式】

例) < 文字列変数 >=C_User

【文例】
```
1 C1$=C_User              'C1$ にはパラメータ「USERMSG」に設定してある文字が入ります。
```

【解説】
(1) パラメータ "USERMSG" に設定されているデータを返します。
(2) 読み込み専用です。
(3) パラメータの変更はパソコンサポートソフトウエアまたは T/B にておこなってください。

J_Curr

【機能】
現在位置の関節型のデータを返します。

【書式】

例) < 関節型変数 >=J_Curr[(< メカ番号 >)]

【用語】
< 関節型変数 > 代入する関節型変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】
```
1 J1=J_Curr                    'J1 には現在の関節位置が入ります。
```

【解説】
(1) 指定したメカ番号の現在位置の関節型データが入ります。
(2) 読み込み専用です。

【関連システム状態変数】
P_Curr

J_ColMxl

【機能】
衝突検知機能有効中の推定トルクと実トルクの差の最大値を返します。

【書式】

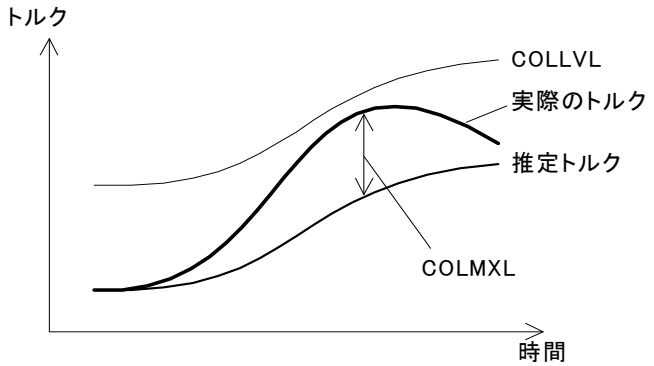
例) < 関節型変数 >=J_ColMxl[(< メカ番号 >)]

【用語】
< 関節型変数 > 代入する関節型変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 M1=100                          ' 各軸の衝突許容レベルの初期値を設定
2 M2=100
3 M3=100
4 M4=100
5 M5=100
6 M6=100
7 *LBL
8 ColLvl M1,M2,M3,M4,M5,M6,, ' 各軸の衝突許容レベルを設定
9 ColChk On                       ' 衝突検知を有効 ( トルク誤差最大値の計算開始 )
10 Mov P1
     ・
     ・
50 ColChk Off                     ' 衝突検知を無効 ( トルク誤差最大値の計算終了 )
51 M1=J_ColMxl(1).J1+10           ' 各軸とも 10% の余裕を見た衝突許容レベルを計算
52 M2=J_ColMxl(1).J2+10             (10% は文例のための参考値で、実際の保証値ではありません )
53 M3=J_ColMxl(1).J3+10
54 M4=J_ColMxl(1).J4+10
55 M5=J_ColMxl(1).J5+10
56 M6=J_ColMxl(1).J6+10
57 GoTo *LBL
```

【解説】
(1) ColChk On ～ Off までの動作を通じ、各軸の推定トルクと実際のトルクの誤差最大値を保持します。

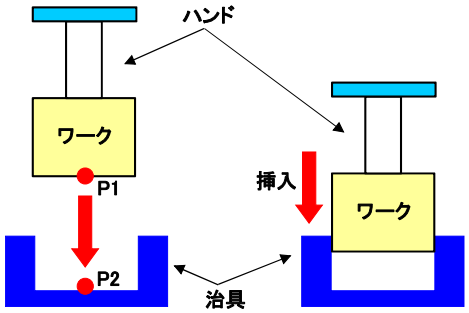


(2) この値が 100% のときは誤差最大値が衝突許容レベルのメーカー初期値と同じであることを表します。
(3) 衝突検知の使用を禁止しているロボットは、常に全軸 0.0 が返されます。
(4) ColChk On、ColLvl 命令実行時、サーボ ON 時に誤差最大値は 0.0 に初期化されます。
(5) 関節型の変数であるため、関節変数としてそのまま読み出すと rad → deg 変換された値となります。使用する際は文例のように各軸要素を数値変数に代入してから使用してください。
(6) 読み込み専用です。

【関連命令】
ColChk（コルチェック）、ColLvl（コルレベル）

【関連システム状態変数】
M_ColSts、P_ColDir

【衝突検知レベルを自動設定するサンプルプログラム】
以下に衝突検知レベルを自動的設定するプログラムを記載します。



左図のように開始点 P1 から終了点 P2 に移動する間の衝突検知レベルを自動設定します。

サンプルプログラム	解説（コメント含む）
'********** 衝突検知レベル自動設定 ********** 'GoSub *LEVEL ' 衝突検知レベル自動設定プログラム 'HLT '** *MAIN Oadl ON LoadSet 2,2 Collvl M_01,M_02,M_03,M_04,M_05,M_06,, Mov PHOME Mov P1 Dly 0.5 ColChk ON Mvs P2 Dly 0.5 ColChk OFF Mov PHOME End	衝突検知レベル自動設定サブルーチンを実行させる命令です。自動設定を行う時だけ先頭のコメントアウトをはずして使用します。 最適加減速にて動作。 ハンドとワークの情報を読み込む。（加減速時間の最適化） 衝突検知レベルを再設定する。 PHOME（待機点）に移動。 P1（開始点）に移動。 衝突検知を有効にします。 衝突検知を無効にします。 PHOME（待機点）に戻ります。 プログラムの終了行です。

サンプルプログラム	解説（コメント含む）
'********** LEVEL FIX ********** *LEVEL Mov PHOME	衝突検知レベル自動調整サブルーチン。
M1=0 'J1 軸の衝突許容レベル（初期化） M2=0 'J2 軸の衝突許容レベル（初期化） M3=0 'J3 軸の衝突許容レベル（初期化） M4=0 'J4 軸の衝突許容レベル（初期化） M5=0 'J5 軸の衝突許容レベル（初期化） M6=0 'J6 軸の衝突許容レベル（初期化） ColLvl 500,500,500,500,500,500,, ' 衝突検知レベルを設定（500%）	J1 軸から J6 軸まで衝突検知レベルを 500%（最高値）に設定。 （動かすときには経路上に障害物がない事を確認する）
For MCHK=1 To 10 Dly 0.3 Mov P1 Dly 0.3 Colhk ON ' 衝突検知有効 Mvs P2 Dly 0.3 ColChk OFF ' 衝突検知無効 If M1<J_ColMxl(1).J1 Then M1=J_ColMxl(1).J1 If M2<J_ColMxl(1).J2 Then M2=J_ColMxl(1).J2 If M3<J_ColMxl(1).J3 Then M3=J_ColMxl(1).J3 If M4<J_ColMxl(1).J4 Then M4=J_ColMxl(1).J4 If M5<J_ColMxl(1).J5 Then M5=J_ColMxl(1).J5 If M6<J_ColMxl(1).J6 Then M6=J_ColMxl(1).J6 Next MCHK	衝突検知レベルを自動で検出できますが、ばらつき等を考慮して複数回実行してください。ここでの繰り返し回数 10 回は参考値です。 J_COLMXL：衝突検知有効中の推定トルクと実トルクの誤差の最大値 ばらつき考慮のため 10 回計測したうち最大値を記憶するようにします。
M_01=M1+10 M_02=M2+10 M_03=M3+10 M_04=M4+10 M_05=M5+10 M_06=M6+10 ColLvl M_01,M_02,M_03,M_04,M_05,M_06,, Mvs P1 Mov PHOME RETURN '************************************	通常電源 ON 後、検知レベルはパラメータ「ColLvl」の値が代入されます。そのため自動設定した値をプログラム外部変数に記録しておきます。 ここで『+10』としているのは自動検知で探した値に対し 10% のマージンを持たせているためです。 ※10% はあくまで参考値であり、ご使用になられているシステムによっては正常に動作しない可能性があります。実機にてご確認の上、最適な調整を行うことを推奨いたします。164 ページの「4.13 命令の詳細説明」の、ColLvl（コルレベル）を参照願います。

J_ECurr

【機能】
現在のエンコーダパルス値を返します。

【書式】

例)<関節型変数>=J_ECurr[(<メカ番号>)]

【用語】
<関節型変数>　代入する関節型変数指定します。(パルス値ですが関節型変数を利用します。)
<メカ番号>　メカ番号を入れます。1～3、省略時は1です。

【文例】
```
1 JA=J_ECurr(1)          'JAにメカ1のエンコーダパルス値が入ります。
2 MA=JA.J1               'MAにJ1軸のエンコーダパルス値を取り出します。
```

【解説】
(1)返される値はパルス値ですが代入する型は関節型を使用してください。その後、関節成分データを指定し、数値変数に代入して利用してください。
(2)読み込み専用です。

J_Fbc／J_AmpFbc

【機能】
J_Fbc：エンコーダのフィードバックより生成された関節型の現在位置を返します。
J_AmpFbc：各軸の現在の電流フィードバック値を返します。

【書式】

例）<関節型変数>=J_Fbc[(<メカ番号>)]

例）<関節型変数>=J_AmpFbc[(<メカ番号>)]

【用語】
<関節型変数>　代入する関節型変数を指定します。
<メカ番号>　メカ番号を入れます。1～3、省略時は1です。

【文例】
1 J1=J_Fbc　'J1には現在サーボのフィードバック値より生成された関節の現在位置が入ります。
2 J2=J_AmpFbc　'J2には現在電流フィードバック値が入ります。

【解説】
(1) J_Fbcはエンコーダのフィードバックより生成された関節型の現在位置を返します。
(2) J_Fbcはサーボへの指令値と実際のサーボの遅れの差が確認できます。
(3) J_FbcはCmp Jnt命令にて差がある場合にも確認できます。
(4) 読み込み専用です。

【関連システム状態変数】
P_Fbc

J_Origin

【機能】
原点設定したときの関節データを返します。

【書式】

例）<関節型変数>=J_Origin[(<メカ番号>)]

【用語】
<関節型変数>　代入する関節型変数を指定します。
<メカ番号>　メカ番号を入れます。1～3、省略時は1です。

【文例】
1 J1=J_Origin(1)　'J1にはメカ1の原点設定位置 が入ります。

【解説】
(1) 原点設定したときの関節データを返します。
(2) ロボットが位置ズレした場合など原点位置の確認に使用します。
(3) 読み込み専用です。

M_Acl/M_DAcl/M_NAcl/M_NDAcl/M_AclSts

【機能】

加減速時間に関する情報を返します。
M_Acl：現在の加速時間比率を返します。(%)
M_DAcl：現在の減速時間比率を返します。(%)
M_NAcl：加速時間の初期値を返します。(100%)
M_NDAcl：減速時間の初期値を返します。(100%)
M_AclSts：現在の加減速状態を返します。
（現在の加減速状態 0= 停止中 1= 加速中 2= 定速中 3= 減速中）

【書式】

例）<数値変数>=M_Acl[(<数式>)]

例）<数値変数>=M_DAcl[(<数式>)]

例）<数値変数>=M_NAcl[(<数式>)]

例）<数値変数>=M_NDAcl[(<数式>)]

例）<数値変数>=M_AclSts[(<数式>)]

【用語】

<数値変数>　　代入する数値変数を指定します。
<数式>　　1 ～ 32、タスクスロット番号を入れます。省略時は現在のスロットです。

【文例】

```
1 M1=M_Acl          'M1 にはタスクスロット 1 で設定されている加速時間比率 が入ります。
2 M1=M_DAcl(2)      'M1 にはタスクスロット 2 で設定されている減速時間比率 が入ります。
3 M1=M_NAcl         'M1 にはタスクスロット 1 で設定されている加速時間初期値比率 が入ります。
4 M1=M_NDAcl(2)     'M1 にはタスクスロット 2 で設定されている減速時間初期値比率 が入ります。
5 M1=M_AclSts(3)    'M1 にはタスクスロット 3 での現在の加減速状態 が入ります。
```

【解説】

(1) 加減速時間比率は個々のロボット最高加減速時間（初期値）に対する比率です。50% の場合は加減速時間が倍になりゆっくりした加減速を行うという意味になります。
(2) M_NAcl, M_NDAcl は常に 100(%) を返します。
(3) 読み込み専用です。

M_BsNo

【機能】
現在のベース座標番号を返します。

【書式】

例) < 数値変数 >=M_BsNo[(< メカ番号 >)]

【用語】

< 数値変数 >	代入する数値変数を指定します。
< メカ番号 >	1 ～ 3 のメカ番号を指定します。(省略時は 1)
	定数、変数、論理 / 算術式、関数が使用可能です。
	実数 / 倍精度実数が指定された場合は小数点以下を四捨五入します。

【文例】

```
1 M1=M_BsNo        ' メカ番号 1 のベース座標番号を変数 M1 へ代入します。
2 If M1=1 Then     ' ベース座標番号が 1 の場合は P1 へ移動します。
3 Mov P1
4 Else             ' ベース座標番号が 1 以外の場合は P2 へ移動します。
5 Mov P2
6 EndIf
```

【解説】
(1) 現在設定されているベース座標番号 (パラメータ : MEXBSNO) を読み出します。
(2) 読み出した値によって以下の座標系が設定されています。
a) 0 : システム初期値 (P_Nbase)
b) 1 ～ 8 : ワーク座標系番号 1 ～ 8 (パラメータ : WK1CORD ～ WK8CORD)
c) -1 : ベース変換が上記以外で設定されています。
(Base 命令でベース変換データを指定, またはパラメータ : MEXBS を直接編集)

【関連命令】
Base (ベース)

【関連パラメータ】
MEXBSNO、WKnCORD(n は 1 ～ 8)、MEXBS

M_BrkCq

【機能】
最後に実行した Break コマンドを含んだ行の実行を行った結果を返します。
1：Break した場合
0：Break しなかった場合

【書式】

例）<数値変数>=M_BrkCq[(<数式>)]

【用語】
<数値変数>　　代入する数値変数を指定します。
<数式>　　1 ～ 32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】
```
1 While M1<>0                    '
2   If M2=0 Then Break           '
3 WEnd
4 If M_BrkCq=1 Then Hlt          'While 内の Break が実行されたら Hlt する。
```

【解説】
(1) Break 命令が実行されたかの状態をチェックします。
(2) 読み込み専用です。
(3) M_BrkCq 変数を一度でも参照すると Break 状態がクリアされます（値がゼロになります）。よって、状態を残したい場合は数値変数へ代入して保存してください。
(4) T/B のモニタ画面等で参照しても Break 状態はクリアされます。

M_BTime

【機能】
バッテリ残容量時間（単位：時間）を返します。

【書式】

例）<数値変数>=M_BTime

【用語】
<数値変数>　　代入する数値変数を指定します。

【文例】
```
1 M1=M_BTime              'M1 にはバッテリ残容量時間が入ります。
```

【解説】
(1) 現時点でバッテリが保つ残り時間を返します。
(2) バッテリの保つ時間の初期値は機種によって異なります。
(3) コントローラの電源を OFF している時間を積算しバッテリ保持時間初期値 14600 から減算した時間を返します。
(4) 読み込み専用です。

M_CavSts

【機能】

干渉を検出したロボット CPU 番号を返します。
1 ～ 3：干渉チェック検出中
0：干渉チェック未検出

本機能は対応機種を限定しています。詳細については「5.24 干渉回避機能について」を参照してください。

【書式】

例) Def Act 1,M_CavSts[(< メカ番号 >)]<>0 GoTo *LCAV,S

【用語】

< メカ番号 >　　メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

500 ページの「5.24.10 サンプルプログラム」を参照。

【解説】

(1) ロボット同士の干渉チェックを検出したときに、干渉を検出したロボット CPU 番号 (1 ～ 3) が書き込まれます。自由平面リミットとの干渉チェックを検出したときは 1 が書き込まれます。
(2) End 命令実行までに自動的に 0 にクリアされないため、割り込み処理で 0 にクリアする必要があります。
(3) End 命令実行またはプログラムリセット、または CavChk On (キャブチェック オン) 命令で "CavChk Off"(干渉回避の停止機能を無効にする) を実行するまで値は保持されます。
(4) NoErr モードで使用する場合の Def Act 命令で割り込み条件に使用します。
(5) 参照および書き込みできます。

M_CmpDst

【機能】
コンプライアンス機能を実行しているときの指令値と実際のロボットのずれ量 (mm) を返します。

【書式】

例) < 数値変数 >=M_CmpDst[(< メカ番号 >)]

【用語】
< 数値変数 > 代入する数値変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 Mov P1
2 CmpG 0.5,0.5,1.0,0.5,0.5, , ,     ' 柔らかさの設定
3 Cmp Pos, &B00011011               ' 柔らかい状態にする
4 Mvs P2
5 M_Out(10)=1
6 Mvs P1
7 M1=M_CmpDst(1)                    ' M1 には動作命令の指令位置と実際の現在位置の差が入ります。
8 Cmp Off                           ' 通常状態に戻す。
```

【解説】
(1) コンプライアンス機能を実行しているときの位置のずれ量を確認するときに使用します。
(2) 読み込み専用です。

M_CmpLmt

【機能】

コンプライアンス機能を実行しているときの指令値が各種リミットを越えようとしているかどうかを返します。
1：リミットを越えようとしている
0：リミットを越えようとしていない。

【書式】

例）Def Act 1, M_CmpLmt [(< メカ番号 >)]=1 GoTo *LMT

【用語】

< メカ番号 >　　メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 Def Act 1, M_CmpLmt(1)=1 GoTo *LMT' 割り込み 1 の条件として定義
2 '
3 '

10 Mov P1
11 CmpG 1,1,0,1,1,1,1,1
12 Cmp Pos, &B100       ' コンプライアンスモードを有効にする
13 Act 1=1              ' 割り込み 1 を有効にする
14 Mvs P2               '
15 '

100 *LMT
101 Mvs P1              ' P2 への動作を中断して P1 へ戻る
102 Reset Err           ' 発生中のエラーをリセット
103 Hlt                 ' 運転を中断する
```

【解説】

(1) コンプライアンスモードの指令値がリミットを越えようとしてエラーとなったとき、割り込み処理を使ってエラー状態から復旧させる場合に使用します。
(2) 各種リミットとして、コンプライアンスモードの指令値の関節動作範囲と動作速度、指令位置と実際の位置ずれ量を確認します。
(3) サーボ電源が OFF していたり、コンプライアンスモードが無効の場合は、0 となります。
(4) 読み込み専用です。

M_ColSts

【機能】

衝突検知の状態を返します。
1：衝突検出中
0：衝突未検出

【書式】

例）Def Act 1, M_ColSts [(< メカ番号 >)]=1 GoTo *LCOL,S

【用語】

< メカ番号 >　　メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 Def Act 1,M_ColSts(1)=1 GoTo *HOME,S      ' 衝突を検知したときに実行する処理を割込で定義
2 Act 1=1
3 ColChk On,NOErr                ' エラー非発生モードで衝突検知を有効にする
4 Mov P1
5 Mov P2                         '40 ～ 70 行実行中に衝突を検知したら割込処理へジャンプする
6 Mov P3
7 Mov P4
8 Act 1=0
    ・
    ・
100 *HOME                        ' 衝突検知時の割込処理
101 ColChk Off                   ' 衝突検知を無効にする
102 Servo On                     ' サーボ ON する
103 PESC=P_ColDir(1)*(-2)        ' 待避動作の移動量を作成
104 PDST=P_Fbc(1)+PESC           ' 待避位置（衝突した位置 (P_Fbc) に退避移動量を加算）を作成する
105 Mvs PDST                     ' 待避位置へ移動
106 Error 9100                   ' ユーザー定義のローレベルエラーを発生させて運転停止
```

【解説】

(1) 衝突を検出したときに 1 になり、サーボ OFF して衝突状態が解除されると 0 になります。
(2) NOErr モードで使用する場合の Def Act 命令で割り込み条件に使用します。
(3) 読み込み専用です。

M_Cstp

【機能】
プログラムがサイクル停止中かどうかの状態を返します。
1：サイクル停止が入力され、サイクル停止運転実効中
（操作パネルの END キー入力、あるいはサイクル停止信号入力）
0：上記以外

【書式】

例）<数値変数>=M_Cstp

【用語】
<数値変数>　　代入する数値変数を指定します。

【文例】
1 M1=M_Cstp　　'M1 には 1 が入ります。（サイクル停止中の場合）

【解説】
(1) プログラムを連続運転している状態で、操作パネルから END キーを押すとサイクル停止状態になります。このときの状態を 1 として返します。
(2) 読み込み専用です。

M_Cys

【機能】
プログラムがサイクル運転中かの状態を返します。
1：サイクル運転中（スロットパラメータ SLT* で運転モード～・・・）
0：上記以外

【書式】

例）<数値変数>=M_Cys

【用語】
<数値変数>　　代入する数値変数を指定します。

【文例】
1 M1=M_Cys　　'M1 には 1 が入ります。（サイクル運転中の場合）

【解説】
(1) プログラムを起動する時、パラメータ等で連続運転かサイクル運転かの運転モードを設定することができます。この運転モードを返します。
(2) スロットパラメータが CYC 指定でも、XRun で連続運転を指定した場合は値が 0 となります。
(3) 読み込み専用です。

M_DIn/M_DOut

【機能】

CR7xx-Q シリーズでは、この状態変数は使用できません。
CC-Link(オプション)のリモートレジスタへの書き込みまたは参照するときに使用します。
M_DIn：入力レジスタの値を参照します。
M_DOut：出力レジスタへ書き込みまたは参照します。

【書式】

例)<数値変数>=M_DIn(<数式 1>)

例)<数値変数>=M_DOut(<数式 2>)

【用語】

<数値変数>	CC-Link のレジスタの値を代入する数値変数を指定します。
<数式 1>	CC-Link のレジスタ番号を指定します。(6000 ～)
<数式 2>	CC-Link のレジスタ番号を指定します。(6000 ～)

【文例】

```
1 M1=M_DIn(6000)        'M1 には CC-Link の入力レジスタ 6000 番の値が入ります。
2 M1=M_DOut(6000)       'M1 には CC-Link の出力レジスタ 6000 番の値が入ります。
3 M_DOut(6000)=100      'CC-Link 出力レジスタ 6000 番へ 100 を書き込みます。
```

【解説】

(1) M_DIn は読み込み専用です。

M_DIn32

【機能】

外部機器から、CC-Link のレジスタを経由して、32 ビットデータを取得します。

【書式】

例)< 数値変数 >=M_DIn32(< 数式 / 入力レジスタ番号 >)

【用語】

< 数値変数 >	代入する長精度整数の数値変数を指定します。
< 数式 / 入力レジスタ番号 >	指定可能な CC-Link のレジスタ番号の範囲は、6000 ～ 6254 です。

【文例】

```
1 M1&=M_DIn32(6000)              'CC-Link のレジスタ番号 6000 と 6001 から 32bit 分の値を取り
                                   出し、その値が M1& に入ります。
2 M2%=6002                       'M2% に 6002 を入れます。
3 M3&=M_DIn32(M2%) And &H7FFFF   'M2% で指定された CC-Link レジスタ番号から 32bit 分の値を取
                                   り出し、その値の下位 19bit 分の値が M3& に入ります。
```

【解説】

(1) 指定した CC-Link レジスタ番号から 2 点を使って、32bit データを取得します。
(2) < 数値変数 > には、長精度整数用の変数をご使用ください。それ以外の場合は、情報が正しく取得できません。
(3) CC-Link のレジスタ番号は、6000 ～ 6254 になるように指定してください。範囲外を指定した場合は、オールゼロを返し、エラーは発生しません。
(4) M_DIn32 は読み込み専用です。

M_DOut32

【機能】

CC-Link のレジスタを経由して、32 ビットデータを外部機器へ出力します。または、現在の出力情報を確認します。

【書式】

参照
＜数値変数＞=M_DOut32(＜数式／出力レジスタ番号＞)

書込み
M_DOut32(＜数式／出力レジスタ番号＞)=＜数値＞

【用語】

＜数値変数(長精度整数)＞	代入する長精度整数の数値変数を指定します。
＜数式／入力レジスタ番号＞	指定可能な CC-Link のレジスタ番号の範囲は、6000 ～ 6254 です。
＜数値＞	出力するデータを指定します。 数値の範囲は、-2147483648 ～ 2147483647(&H80000000 ～ &H7FFFFFFF)となります。

【文例】

```
1 M_DOut32(6000)=&H12345678        'CC-Link のレジスタ番号 6000 と 6001 に 16 進数表記の
                                    12345678 を出力します。
                                    レジスタ番号 6000 に＆H4567、6001 に &H1234 が出力されます。
2 M1&=M_DOut32(6002) And &H7FFFF   'CC-Link レジスタ番号 6002 と 6003 に出力されている 32bit
                                    データから、下位 19 ビット分の値が M1& に入ります。
```

【解説】

(1) 指定した CC-Link レジスタ番号から 2 点に対して、指定した 32bit データを出力します。
(2) 指定した CC-Link レジスタ番号から 2 点に対して、現時点の出力情報(32bit データ)を取得します。
(3) ＜数値変数＞と＜数値＞には、長精度整数用の変数をご使用ください。それ以外の場合は、情報が扱えない場合があります。
(4) CC-Link のレジスタ番号は、6000 ～ 6254 になるように指定してください。範囲外を指定した場合、参照時はオールゼロが返り、書き込み時は無処理でエラーは発生しません。ただし、マイナス値を指定した場合は、L3110 エラーが発生します。
(5) パラメータ SYNCIO を高速モードに設定することにより、レジスタ出力の更新周期を速くすることができます。パフォーマンス向上のために信号のインターロックをおこなったうえで、高速モードでご使用いただくことをお勧めします。
(6) 指定されたレジスタ番号がロボットが占有している番号でない場合、RT ToolBox2 などのレジスタ出力モニタ上では出力されている情報となりますが、実際には CC-Link を通じて出力されません。また、占有している最後のレジスタを指定した場合も、モニタ上では出力されていますが、実際には占有している最後のレジスタの情報(下位 16bit 分)しか出力されません。
(7) Dly 命令を組み合わせてのパルス出力には対応していません。Dly 命令を使用すると、L4220(入力した命令文の構文に誤りがあります)が発生します。
(8) パラメータ DIODATA および SVDATA の専用出力に割り当てられたレジスタ番号への出力時は、H0091(この信号は専用出力に割り付け済みです)エラーが発生します。

M_ErCode

【機能】
現在発生しているエラーの詳細エラー番号を返します。

【書式】

例) < 数値変数 >=M_ErCode

【用語】
< 数値変数 > 32 ビット長精度整数を指定します。
(16 ビット単精度整数では代入時に範囲外エラーになります)
読み出される値の意味は【解説】(2) のとおりです。

【文例】
```
1 *LBL: If M_Err=0 Then *LBL        ' エラーが発生するまで待ちます。
2 MD&=M_ErCode                      ' 詳細エラー番号を読み出します ( 長精度整数に代入 )
3 MS%=Int(MD&/1000) Mod 100         ' 詳細エラー番号の 2 桁部分を取り出します。
```

【解説】
(1) 複数のエラーが発生した場合は、もっともエラーレベルの高い情報を返します。(1 つのみ)
(2) 詳細エラー番号は最大 9 桁の番号です。
XXXXYYYYY.....XXXX : O/P や T/B で表示される 4 桁のエラー番号
YYYYY : 詳細エラー番号 (内容はエラーによって決められています)

M Err/M ErrLvl/M ErrNo

【機能】

ロボットのエラーに関する情報を返します。

M_Err：エラー発生中かどうか

M_ErrLvl：発生しているエラーのレベル

M_ErrNo：発生しているエラー番号

【書式】

例）<数値変数>=M_Err

例）<数値変数>=M_ErrLvl

例）<数値変数>=M_ErrNo

【用語】

<数値変数>　　代入する数値変数を指定します。

代入される値と意味

M_Err　：0/1= エラー発生なし / エラー発生中

M_ErrLvl　：0/1/2/3/4/5/6= エラー発生なし / 警告 / ﾛｰﾚﾍﾞﾙｴﾗｰ / ﾊｲﾚﾍﾞﾙｴﾗｰ / 警告 1/ ﾛｰﾚﾍﾞﾙｴﾗｰ 1/ ﾊｲﾚﾍﾞﾙｴﾗｰ 1

注）各用語と意味を表 4-18 に示します。

M_ErrNo　：発生しているエラーの番号

【文例】

```
1 *LBL: If M_Err=0 Then *LBL        ' エラーが発生するまで待ちます。
2 M2=M_ErrLvl                       ' M2 にエラーのレベルが入ります。
3 M3=M_ErrNo                        ' M3 にエラー番号が入ります。
```

【解説】

(1) 通常のプログラムはエラー(caution 以外)が発生すると中断しますが、SLT* パラメータにて起動条件が常時実行(ALWAYS)に設定してあるプログラムにてこの変数を利用してコントローラのエラー状態を監視することが可能です。常時実行プログラムは他のプログラムでエラーがでても停止しません。

(2) 複数のエラーが発生した場合は、最もエラーレベルの高いエラーの情報を返します。

(3) M_ErrLvl が返すエラーレベルとその意味を以下に示します。

表 4-18：エラーレベルと意味

エラーレベル	用語	意味	エラーリセット
0	エラーなし	エラーは発生していません	-
1	警告	警告でプログラムは継続します	[RESET] キー
2	ﾛｰﾚﾍﾞﾙｴﾗｰ	実行中のプログラムは中断します	[RESET] キー
3	ﾊｲﾚﾍﾞﾙｴﾗｰ	実行中のプログラムは中断し、サーボ電源を切ります	[RESET] キー
4	警告 1	警告でプログラムは継続します	電源リセット
5	ﾛｰﾚﾍﾞﾙｴﾗｰ 1	実行中のプログラムは中断します	電源リセット
6	ﾊｲﾚﾍﾞﾙｴﾗｰ 1	実行中のプログラムは中断し、サーボ電源を切ります	電源リセット

【関連命令】

Error（エラー）、Reset Err(リセット エラー)

M_ESpd

【機能】
Ex-T 制御 /Ex-T スプライン補間命令実行時の Ex-T 座標通過速度 [mm/s] を返します。

【書式】

例) < 数値変数 > = M_ESpd [(< メカ番号 >)]

【用語】
< 数値変数 > 代入する数値変数を指定します。
< メカ番号 > Ex-T 制御 /Ex-T スプライン補間を実行しているメカ番号を設定します。
設定範囲 : 1 ～ 3
省略時 : 1
存在しないメカの番号を指定した場合は、実行時にエラー L3870（指定したメカ番号が無効な値です）が発生します。

【文例】
```
1 Wait M_00=1            'Ex-T スプライン補間が開始するのを待つ
2 *L1:If M_ESpd<50 Then GoTo *L1
3 M_Out(100)=1           'Ex-T 座標通過速度が 50[mm/s] 以上になった場合、出力 100 を ON
```

【解説】
(1) Ex-T 制御 /Ex-T スプライン補間命令実行中の Ex-T 座標通過速度 [mm/s] を返します。
(2) M_ESpd の値をマルチタスクで参照することにより、Ex-T 座標通過速度に応じて演算処理や信号出力等の処理を実行することができるようになります。
(3) Ex-T 制御 /Ex-T スプライン補間命令を実行していない場合は、0.0 を返します。
(4) M_ESpd は状況に応じて下表に示す値を返します。
表 4-19 : M_ESpd が返す値

状況	M_ESpd が返す値
電源投入直後	0.0
Ex-T 制御 /Ex-T スプライン補間実行中	補間実行時の Ex-T 座標通過速度
Ex-T 制御 /Ex-T スプライン補間以外の補間実行中	0.0
補間中断時	0.0
メインプログラムの End 命令実行直後	0.0
プログラムリセット操作直後	0.0
スプライン補間に対応していないメカ	0(書き込み実行後も 0)

(5) 読み込み専用です。

【関連命令】
EMvc（イームーブシー）、EMvr（イームーブアール）、EMvr2（イームーブアール 2）、EMvr3（イームーブアール 3）、EMvs（イームーブエス）、EMvSpl（イームーブスプライン）

M_Exp

【機能】
自然対数の底 (2.71828182845905) を返します。

【書式】

例) < 数値変数 >=M_Exp

【用語】
< 数値変数 > 代入する数値変数を指定します。

【文例】
```
1 M1=M_Exp          'M1 には自然対数の底 (2.71828182845905) が入ります。
```

【解説】
(1) 指数関数、対数関数処理をする場合に利用します。
(2) 読み込み専用です。

M_Fbd

【機能】
指令位置とフィードバック位置の差を返します。

【書式】

例) < 数値変数 >=M_Fbd [(< メカ番号 >)]

【用語】
< 数値変数 > 代入する数値変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】
```
1 Def Act 1,M_Fbd>10 GoTo *SUB1,S  ' 指令位置とフィードバック位置の差が 10mm 以上で割込み
2 Act 1=1                          ' 割り込み有効
3 Torq 3,10                        ' トルク命令で 3 軸を 10% 以下に設定
4 Mvs P1                           ' 動作
5 End

10 *SUB1
11 Mov P_Fbc                       ' 指令位置とフィードバック位置を一致させる
12 M_Out(10)=1                     ' 信号 10 番出力
13 End                             ' 差がでた時点で停止。
```

【解説】
(1) 動作命令での指令位置とモータからのフィードバック位置との差を返す関数です。
差は、ロボット本体の XYZ 軸の合成値です。(走行軸の差は含まれていません)
トルク命令などを使用している場合で誤差過大エラー (960,970 など) でエラーにしたくない時に Def Act 命令と組み合わせてお使いください。
(2) 読み込み専用です。

【関連命令】
Torq (トルク)

【関連システム状態変数】
P_Fbc

M_G

【機能】
重力定数 (9.80665) を返します。

【書式】

例) < 数値変数 >=M_G

【用語】
< 数値変数 > 代入する数値変数を指定します。

【文例】
1 M1=M_G 'M1 には重力定数 (9.80665) が入ります。

【解説】
(1) 重力に関する演算をする場合に利用します。
(2) 読み込み専用です。

M_HndCq

【機能】
ハンドチェック入力信号の値を返します。

【書式】

例) < 数値変数 >=M_HndCq(< 数式 >)

【用語】
< 数値変数 > 代入する数値変数を指定します。
< 数式 > ハンド入力信号番号を入れます。
1 ～ 8(入力信号 900 から 907 に相当します)。

【文例】
1 M1=M_HndCq(1) 'M1 にはハンド 1 の状態が入ります。

【解説】
(1) ハンドチェック入力信号状態 (センサなど) の 1 ビット分を返します。
(2) M_HndCq (1) は入力信号 900 番に相当します。M_In(900) を使用しても同じ結果になります。
(3) 読み込み専用です。

M_In/M_Inb/M_In8/M_Inw/M_In16

【機能】

入力信号の値を返します。
M_In：ビット信号を返します。
M_Inb または M_In8：バイト単位（8 ビット分）で返します。
M_Inw または M_In16：ワード単位（16 ビット分）で返します。

【書式】

例）<数値変数>=M_In(<数式>)

例）<数値変数>=M_Inb(<数式>) または M_In8(<数式>)

例）<数値変数>=M_Inw(<数式>) または M_In16(<数式>)

【用語】

<数値変数>　代入する数値変数を指定します。表 4-20 に補足内容を示します。
<数式>　信号番号を指定します。表 4-21 に補足内容を示します。
(1)CR7xx-Q シリーズ
10000 ～ 18191：マルチ CPU 間共有デバイス
716 ～ 731：マルチハンド
900 ～ 907：ハンド
(2)CR7xx-D シリーズ
0 ～ 255：パラレル入出力
716 ～ 731：マルチハンド
900 ～ 907：ハンド
2000 ～ 5071：プロフィバス
6000 ～ 8047：CC-Link

【文例】

```
1 M1%=M_In(10010)                ’M1 には入力信号 10010 番の値（1 か 0）が入ります。
2 M2%=M_Inb(900)                 ’M2 には入力信号 900 番から 8 ビット分の値が入ります。
3 M3%=M_Inb(10300) And &H7       ’M3 には入力信号 10300 番から 3 ビット分の値が入ります。
4 M4%=M_Inw(15000)               ’M4 には入力信号 15000 番から 16 ビット分の値が入ります。
```

【解説】

(1) 入力信号の状態を返します。
(2) 指定した信号番号から M_In は 1 ビット、M_Inb/M_In8 は 8 ビット、M_Inw/M_In16 は 16 ビット分のデータをそれぞれ返します。
(3) 信号番号には 32767 まで入りますが、実際にハードウエアが存在する信号番号のみ値が返ります。ハードウエアが存在しない信号番号の場合は処理をしません。
(4) 読み込み専用です。

⚠注意　ロボットと周辺装置間で信号を確実に受け渡すために、信号のインタロックは必ず実施するようにしてください。インターロックをおこなわないと信号が正しく伝わらず誤動作の原因となります。

【補足】

表 4-20 :＜数値変数＞に記述できる変数

○：使用可能、×：使用不可能

ビット幅	数値変数の型				その他の変数		
	整数 例)M1%	長精度整数 例)M1&	単精度 実数 例)M1!	倍精度 実数 例)M1#	位置注 1) 例)P1.X	関節注 1) 例)J1.J1	文字列 例)C1$
M_In	○	○	○	○	○	○	×
M_Inb/M_In8	○	○	○	○	○	○	×
M_Inw/M_In16	○	○	○	○	○	○	×

注 1）変数で扱う数値が角度の場合（位置変数の成分データ A,B,C と関節変数の全成分データ）、入力信号をラジアン単位の数値として格納するため、信号のオン / オフ状態が明確に把握できなくなります。（モニタなどの表示では度の単位に変換して表示します）
例)P1.A=M_In(8) で入力信号 8 がオンの時、P1.A は "57.296" と表示され信号のオン / オフ状態が明確に把握できません。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

表 4-21 :＜数式＞に記述できる定数・変数

○：使用可能、×：使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値注 1) 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度 実数注 1) 例)M1!	倍精度 実数注 1) 例)M1#	位置変数 注 1) 注 2) 例)P1.X	関節変数 注 1) 注 2) 例)J1.J1	文字列 例)C1$
M_In	○	○	○	○	○	○	○	○	○	×
M_Inb/M_In8	○	○	○	○	○	○	○	○	○	×
M_Inw/M_In16	○	○	○	○	○	○	○	○	○	×

注 1）実数値は四捨五入されます。
注 2）変数で扱う数値が角度の場合（位置変数の成分データ A,B,C と関節変数の全成分データ）、設定した値をラジアン単位の数値で格納して処理するため、信号番号の指定が難しくなります。（モニタなどの表示では度の単位に変換して表示するので設定値と同じ値を表示します）
例) 入力信号 8 番を指定するため P1.A に 8 を設定 (T/B 等からのキー入力) しても内部処理は 0.14(rad) を四捨五入して 0 となり、入力信号 0 番を指定することになってしまいます。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

【関連システム状態変数】

M_Out32

【関連命令】

Def IO（デファイオー）

M_In32

【機能】
32 ビット幅の外部入力信号を数値として読み出します。

【書式】

例) < 数値変数 >=M_In32(< 数式 >)

【用語】
< 数値変数 > 代入する数値変数を指定します。表 4-22 に補足内容を示します。
< 数式 > 信号番号を指定します。表 4-23 に補足内容を示します。
(1)CR7xx-Q シリーズ
10000 ～ 18191 : マルチ CPU 間共有デバイス
716 ～ 731 : マルチハンド
900 ～ 907 : ハンド
(2)CR7xx-D シリーズ
0 ～ 255 : パラレル入出力
716 ～ 731 : マルチハンド
900 ～ 907 : ハンド
2000 ～ 5071 : プロフィバス
6000 ～ 8047 : CC-Link

【文例】
```
1 *ack_wait
2 If M_In(7)=0 Then *ack_wait   ' 入力信号 7 がオンするまで待つ ( インターロック )
3 M1&=M_In32(10000)             ' 長精度数値変数 M1 に入力信号 10000 から 32 ビット分のデータを数
                                  値として格納する。
4 P1.Y=M_In32(10100)/1000.0     ' 位置変数 P1 の Y 成分に入力信号 10100 から 32 ビット分のデータを
                                  数値として入力し、1/1000 倍して ( 実数化 ) 格納する。
                                  ( 外部機器側ではロボットの位置変数 P1 の成分データ Y に設定す
                                  る数値として、実数値を 1000 倍して整数化しロボットへ出力して
                                  いる時の例です )
```

【解説】
(1) 信号番号から 32 ビット分のデータを数値として返します。
(2) 信号番号には 32767 まで入りますが、実際にハードウエアが存在する信号番号のみ値が返ります。
ハードウエアが存在しない信号番号の場合は処理をしません。
(3) < 数値変数 > には長精度整数型、または実数型の変数を指定してください。
(4) 読み込み専用です。

⚠注意 ロボットと周辺装置間で信号を確実に受け渡すために、信号のインタロックは必ず実施するようにしてください。インターロックをおこなわないと信号が正しく伝わらず誤動作の原因となります。

【補足】
表 4-22 : < 数値変数 > に記述できる変数

○ : 使用可能、× : 使用不可能

ビット幅	数値変数の型				その他の変数		
	整数 例)M1%	長精度整数 例)M1&	単精度 実数 例)M1!	倍精度 実数 例)M1#	位置注 1) 例)P1.X	関節注 1) 例)J1.J1	文字列 例)C1$
M_In32	×	○	○	○	○	○	×

注 1) 変数で扱う数値が角度の場合 (位置変数の成分データ A,B,C と関節変数の全成分データ)、32 ビットの入力データをそのままラジアン単位の数値として格納します。モニタなどでは度の単位に変換して表示します。
例)P1.A=M_In32(16) でその入力データが 5(10 進数) の時、P1.A は
"286.479"(deg) と表示します。

表 4-23：＜数式＞に記述できる定数・変数

○：使用可能、×：使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値 注 1） 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度実数 注 1） 例)M1!	倍精度実数 注 1） 例)M1#	位置 注 1） 注 2） 例)P1.X	関節 注 1） 注 2） 例)J1.J1	文字列 例)C1$
M_In32	○	○	○	○	○	○	○	○	○	×

注 1）実数値は四捨五入されます。

注 2）変数で扱う数値が角度の場合（位置変数の成分データ A,B,C と関節変数の全成分データ）、設定した値をラジアン単位の数値で格納して処理しするため、信号番号の指定が難しくなります。（モニタなどの表示では度の単位に変換して表示するので設定値と同じ値を表示します）

例）入力信号 8 番を指定するため P1.A に 8 を設定（T/B 等からのキー入力）しても内部処理は 0.14(rad)を四捨五入して 0 となり、入力信号 0 番を指定することになってしまいます。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

【関連システム状態変数】

M_In/M_Inb/M_In8/M_Inw/M_In16

【関連命令】

Def IO（デファイオー）

M_JOvrd/M_NJOvrd/M_OPOvrd/M_Ovrd/M_NOvrd

【機能】

オーバーライド値を返します。
M_JOvrd：関節補間用のオーバーライド JOvrd 命令で指定された値
M_NJOvrd：関節補間用のオーバーライドの初期値（100%）
M_OPOvrd：操作パネルのオーバーライド値
M_Ovrd：現在のオーバーライド値、Ovrd 命令で指定された値
M_NOvrd：オーバーライドの初期値（100%）

【書式】

例）＜数値変数＞=M_JOvrd[(＜数式＞)]

例）＜数値変数＞=M_NJOvrd[(＜数式＞)]

例）＜数値変数＞=M_OPOvrd

例）M_OPOvrd=＜数式＞

例）＜数値変数＞=M_Ovrd[(＜数式＞)]

例）＜数値変数＞=M_NOvrd[(＜数式＞)]

【用語】

＜数値変数＞	代入する数値変数を指定します。
＜数式＞	M_JOvrd/M_NJOvrd/M_Ovrd/M_NOvrd の数式には、1 ～ 32、タスクスロット番号を入れます。省略時は現在のスロットです。 M_OPOvrd の数式には、オーバーライド値を入れます。

【文例】

```
1 M1=M_Ovrd          'M1 には現在のオーバーライド値が入ります。
2 M2=M_NOvrd         'M2 には現在のオーバーライドの初期値（100%）が入ります。
3 M3=M_JOvrd         'M3 には現在の関節オーバーライド値が入ります。
4 M4=M_NJOvrd        'M4 には現在の関節オーバーライドの初期値が入ります。
5 M5=M_OPOvrd        'M5 には現在の OP（操作パネル）オーバーライド値が入ります。
6 M6=M_Ovrd(2)       'M6 にはスロット 2 の現在のオーバーライド値が入ります。
7 M_OPOvrd=20        'OP（操作パネル）オーバーライドに 20 が設定されます。
                      注）M_OPOvrd への設定は以下のソフトウェアバージョンで有効です。
                          F-Q シリーズ：R5 以降、F-D シリーズ：S5 以降
```

【解説】

(1) 引数を省略した場合は現在のスロットの状態を返します。
(2) M_JOvrd/M_NJOvrd/M_Ovrd/M_NOvrd は読み出しのみ可能です。
M_OPOvrd は、読み出し、書き込みとも可能です。

M_Line

【機能】
現在実行しているステップ番号を返します。

【書式】

例) < 数値変数 >=M_Line[(< 数式 >)]

【用語】
< 数値変数 > 代入する数値変数を指定します。
< 数式 > 1 ～ 32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】
```
1 M1=M_Line(2)          'M1 スロット 2 の実行ステップ番号が入ります。
```

【解説】
(1) マルチタスク運転時に他のタスクの実行ステップを監視する場合に利用します。
(2) 読み込み専用です。

M_LdFact

【機能】
各関節軸単位の負荷率を参照できます。
【書式】

例）<数値変数>=M_LdFact(<軸番号>)

【用語】
<軸番号>　　軸番号を指定します。(1 ～ 8)
<数値変数>　　各軸の負荷率が代入されます。0 ～ 100(%) です。

【文例】

```
1 Accel 100,100                ' 全体の減速時間を 100% にします。
2 *Label
3 Mov P1
4 Mov P2
5 If M_LdFact(2)>90 Then
6     Accel 50,50              ' 加減速比を 50% に下げる
7     M_SetAdl(2)=50           'J2 軸の加減速比をさらに 50% にする、( 実際は 50%×50%=25%)
8 Else
9     Accel 100,100            ' 加減速時間を戻す
10 EndIf
11 GoTo *Label
```

【解説】
(1) 各軸の負荷率を参照できます。
(2) 負荷率は各軸モータへ流れる電流とその流れる時間から導き出しています。
(3) 高負荷できつい姿勢を長時間動作させると負荷率は上昇します。
(4) 負荷率が 100% になると過負荷エラーが発生します。上記文例では負荷率が 90% をこえた場合は加減速時間を一旦 50% まで下げる例です。
(5) 負荷率を下げるには加減速時間を下げるか、ロボットを無理のない姿勢にて待機させるか、サーボ電源を切るかの対策が有効です。
(6) 対象とするメカ番号の初期値はメカ 1 となっています。RelM(リリースメカ) 命令を実行した後や、マルチタスク制御でスロット 1 以外のプログラムでメカ 1 を対象とする場合、GetM 命令によるメカ番号指定は不要です。他のメカを対象とする場合に GetM 命令を実行してください。

【関連命令】
Accel (アクセル)、Ovrd (オーバーライド)

M_LdFMax

【機能】
各関節軸単位の最大負荷率を参照できます。
【書式】

例)<数値変数>=M_LdFMax(<軸番号>)

【用語】
<軸番号>　　軸番号を指定します。(1 ～ 8)
<数値変数>　　各軸の負荷率が代入されます。0 ～ 100(%) です。

【文例】

```
1 Accel 100,100                 ' 全体の減速時間を 100% にします。
2 *Label
3 Mov P1
4 Mov P2
5 If M_LdFMax(2)>90 Then Goto *LdErr Else Goto *Label
6
7 *LdErr
8 Error 9101                    ' ユーザーエラー出力
9 End
```

【解説】
(1) 各軸の最大負荷率を参照できます。ロボットの動作の厳しさをチェックできます。
(2) 負荷率は各軸モータへ流れる電流とその流れる時間から導き出しています。
(3) 高負荷できつい姿勢を長時間動作させると負荷率は上昇します。
(4) 負荷率が 100% になると過負荷エラーが発生します。上記文例では負荷率が 90% をこえた場合は加減速時間を一旦 50% まで下げる例です。
(5) 負荷率を下げるには加減速時間を下げるか、ロボットを無理のない姿勢にて待機させるか、サーボ電源を切るかの対策が有効です。
(6) 対象とするメカ番号の初期値はメカ 1 となっています。RelM(リリースメカ)命令を実行した後や、マルチタスク制御でスロット 1 以外のプログラムでメカ 1 を対象とする場合、GetM 命令によるメカ番号指定は不要です。他のメカを対象とする場合に GetM 命令を実行してください。

【関連命令】
Accel(アクセル)、Ovrd(オーバーライド)

M_Mode

【機能】
コントローラのモードを返します。
1：MANUAL
2：AUTOMATIC(操作パネル)
3：AUTOMATIC(外部機器)

【書式】

例)<数値変数>=M_Mode

【用語】
<数値変数>　　代入する数値変数を指定します。

【文例】
1 M1=M_Mode　　'M1 には現在のコントローラのモードが入ります。

【解説】
(1) マルチタスク運転時、ALWAYS 実行(常時実行)されているプログラムなどにて利用します。
(2) 読み込み専用です。

M_On/M_Off

【機能】
常に 1(M_On) または 0(M_Off) を返します。

【書式】

例)<数値変数>=M_On

例)<数値変数>=M_Off

【用語】
<数値変数>　　代入する数値変数を指定します。

【文例】
1 M1=M_On　　'M1 には 1 が入ります。
2 M2=M_Off　　'M2 には 0 が入ります。

【解説】
(1) 常に 1 またはゼロを返します。
(2) 読み込み専用です。

M_Open

【機能】

指定されたファイルのオープン状態を返します。Open 命令で指定したファイル種類により返る値が異なります。

【書式】

例) < 数値変数 >=M_Open[(< ファイル番号 >)]

【用語】

< 数値変数 > 代入する数値変数を指定します。

< ファイル番号 > Open 命令でオープンしたファイル番号 1 ～ 8 の定数を指定します。
省略時はファイル番号 1 となります。9 以上は実行時にエラーになります。

【文例】

```
1 Open "temp.txt" As #1                  ' "temp.txt" をファイル番号 1 として OPEN
2 *LBL:If M_Open(1)<>1 Then GoTo *LBL    ' ファイル番号 1 が OPEN するまで待つ
```

【解説】

(1) 読み込み専用です。

(2) Open 命令で指定したファイルの種類により返る値は以下のようになります。

Open 命令実行時の ファイルの種類	意味	値
ファイル	ファイルがオープンされているか、オープンされていないかの状態を返します。 Open 命令実行後、Close 命令、End 命令、プログラムの End が実行されるまで、1 を返します。	1 : オープン済み -1 : ファイル番号未定義 (未 OPEN)

【関連命令】

Open (オープン)

M_Out/M_Outb/M_Out8/M_Outw/M_Out16

【機能】

外部出力信号への書き込み、または参照します。
M_Out：出力信号のビット
M_Outb または M_Out8：出力信号のバイト（8 ビット分）
M_Outw または M_Out16：出力信号のワード（16 ビット分）

【書式】

例）M_Out(< 数式 1>)=< 数値 2>
例）M_Outb(< 数式 1>) または M_Out8(< 数式 1>)=< 数値 3>
例）M_Outw(< 数式 1>) または M_Out16(< 数式 1>)=< 数値 4>
例）M_Out(< 数式 1>)=< 数値 2> Dly < 時間 >
例）< 数値変数 >=M_Out(< 数式 1>)

【用語】

< 数式 1>　出力信号の番号を指定します。表 4-24 に補足内容を示します。
(1)CR7xx-Q シリーズ
10000 ～ 18191：マルチ CPU 間共有デバイス
716 ～ 723：マルチハンド
900 ～ 907：ハンド
(2)CR7xx-D シリーズ
0 ～ 255：パラレル入出力
716 ～ 723：マルチハンド
900 ～ 907：ハンド
2000 ～ 5071：プロフィバス
6000 ～ 8047：CC-Link

< 数値 2>,< 数値 3>,< 数値 4>
出力するデータを数値変数、定数、または数値演算式で記述します。表 4-25 に補足内容を示します。
数値の範囲
< 数値 2>：0 または 1（&H0 または &H1）
< 数値 3>：-128 ～ +127（&H80 ～ &H7F）
< 数値 4>：-32768 ～ +32767（&H8000 ～ &H7FFF）

< 時間 >　パルス出力の出力時間を定数または数値演算式で記述します。単位：[秒]

< 数値変数 >　出力信号を格納する数値変数を指定します。表 4-26 に補足内容を示します。

【文例】

```
1 M_Out(902)=1                ' 出力信号 902(1 ビット ) を ON します。
2 M_Outb(10016)=&HFF          ' 出力信号 10016 から 8 ビット分を ON します。
3 M_Outw(10032)=&HFFFF        ' 出力信号 10032 から 16 ビット分を ON します。
4 M4=M_Outb(10200) And &H0F   ' M4 には出力信号 10200 から下位 4 ビット分の値が入ります。
```

【解説】

(1) 外部出力信号への書き込みまたは参照をするときに使用します。
(2) 指定した信号番号から M_Out は 1 ビット、M_Outb/M_Out8 は 8 ビット、M_Outw/M_Out16 は 16 ビット分のデータをそれぞれ出力します。
(3) パルス出力については、203 ページの「Dly（ディレイ）」を参照してください。
(4) パラメータ：SYNCIO を高速モードに設定することにより、外部出力信号への更新周期を速くすることができます。パフォーマンス向上のために信号のインターロックをおこなった上で、高速モードでご使用いただくことをお勧めします。詳細は 419 ページの「5.2 信号パラメータ」の SYNCIO を参照願います。

⚠注意　ロボットと周辺装置間で信号を確実に受け渡すために、信号のインタロックは必ず実施するようにしてください。インターロックをおこなわないと信号が正しく伝わらず誤動作の原因となります。

【補足】

表 4-24 : < 数式 1> に記述できる定数・変数

○ : 使用可能、× : 使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値注 1) 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度実数注 1) 例)M1!	倍精度実数注 1) 例)M1#	位置 注 1) ,注 2) 例)P1.X	関節 注 1) , 注 2) 例)J1.J1	文字列 例)C1$
M_Out	○	○	○	○	○	○	○	○	×	×
M_Outb/M_Out8	○	○	○	○	○	○	○	○	×	×
M_Outw/M_Out16	○	○	○	○	○	○	○	○	×	×

注 1) 実数値は四捨五入されます。

注 2) 変数で扱う数値が角度の場合 (位置変数の成分データ A,B,C と関節変数の全成分データ)、設定した値をラジアン単位の数値で格納して処理しするため、信号番号の指定が難しくなります。(モニタなどでは度の単位に変換して表示するので設定値と同じ値を表示します)

例) 出力信号 8 番を指定するため P1.A に 8 を代入 (P1.A=8) しても内部処理は 0.14(rad) を四捨五入して 0 となり、出力信号 0 番を指定することになってしまいます。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

表 4-25 : < 数値 2>,< 数値 3>,< 数値 4> に記述できる定数・変数

○ : 使用可能、× : 使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度実数 例)M1!	倍精度実数 例)M1#	位置 例)P1.X	関節 例)J1.J1	文字列 例)C1$
M_Out	○	○	○	○	×	×	×	×	×	×
M_Outb/M_Out8	○	○	○	○	×	×	×	×	×	×
M_Outw/M_Out16	○	○	○	○	×	×	×	×	×	×

表 4-26 : < 数値変数 > に記述できる変数

○ : 使用可能、× : 使用不可能

ビット幅	数値変数の型				その他の変数		
	整数 例)M1%	長精度整数 例)M1&	単精度実数 例)M1!	倍精度実数 例)M1#	位置注 1) 例)P1.X	関節注 1) 例)J1.J1	文字列 例)C1$
M_Out	○	○	○	○	○	○	×
M_Outb/M_Out8	○	○	○	○	○	○	×
M_Outw/M_Out16	○	○	○	○	○	○	×

注 1) 変数で扱う数値が角度の場合 (位置変数の A,B,C 成分データと関節変数の全成分データ)、現在の出力状態をラジアン単位の数値で格納し演算などの処理はラジアン単位で実行します。モニタなどでは度の単位に変換して表示します。

【関連システム状態変数】

M_Out32

【関連命令】

Def IO (デファイオー)

【関連パラメータ】

SYNCIO

M_Out32

【機能】
32 ビット幅の外部出力信号へ数値の書き込み、または参照します。

【書式】

例) M_Out32(< 数式 1>)=< 数値 >
例) < 数値変数 >=M_Out32(< 数式 1>)

【用語】

< 数式 1>　出力信号の番号を指定します。表 4-24 に補足内容を示します。
(1)CR7xx-Q シリーズ
10000 ～ 18191 : マルチ CPU 間共有デバイス
716 ～ 723 : マルチハンド
900 ～ 907 : ハンド
(2)CR7xx-D シリーズ
0 ～ 255 : パラレル入出力
716 ～ 723 : マルチハンド
900 ～ 907 : ハンド
2000 ～ 5071 : プロフィバス
6000 ～ 8047 : CC-Link

< 数値 >　出力するデータを数値変数、定数、または数値演算式で記述します。表 4-25 に補足内容を示します。
数値の範囲 : -2147483648 ～ +2147483647 (&H80000000 ～ &H7FFFFFFF)

< 数値変数 >　出力信号を格納する数値変数を指定します。表 4-26 に補足内容を示します。

【文例】

```
1 M_Out32(10000)=P1.X * 1000    ' 位置変数 P1 の X 座標値を出力信号番号 10000 から 32 ビット分に
                                  1000 倍して ( 整数化 ) 出力する。
2 *ack_wait
3 If M_In(7)=0 Then *ack_wait   ' 入力信号 7 がオンするまで待つ ( インターロック )
4 P1.Y=M_In32(10100)/1000.0     ' 位置変数 P1 の Y 成分に入力信号 10100 から 32 ビット分のデータを
                                  数値として入力し、1/1000 倍して ( 実数化 ) 格納する。
                                  ( 外部機器側ではロボットの位置変数 P1 の成分データ Y に設定す
                                  る数値として、実数値を 1000 倍して整数化しロボットへ出力して
                                  いる時の例です )
```

【解説】
(1) 32 ビット幅の外部出力信号への書き込み、または参照をするときに使用します。
(2) 指定した信号番号から 32 ビット分のデータを出力します。
(3) パラメータ : SYNCIO を高速モードに設定することにより、外部出力信号への更新周期を速くすることができます。パフォーマンス向上のために信号のインターロックをおこなった上で、高速モードでご使用いただくことをお勧めします。詳細は 419 ページの「5.2 信号パラメータ」の SYNCIO を参照願います。

⚠注意　ロボットと周辺装置間で信号を確実に受け渡すために、信号のインタロックは必ず実施するようにしてください。インターロックをおこなわないと信号が正しく伝わらず誤動作の原因となります。

【補足】

表 4-27 : < 数式 1> に記述できる定数・変数

○ : 使用可能、× : 使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値[注 1)] 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度実数[注 1)] 例)M1!	倍精度実数[注 1)] 例)M1#	位置[注 1) ,注 2)] 例)P1.X	関節[注 1) , 注 2)] 例)J1.J1	文字列 例)C1$
M_Out32	○	○	○	○	○	○	○	○	×	×

注 1) 実数値は四捨五入されます。

注 2) 変数で扱う数値が角度の場合 (位置変数の成分データ A,B,C と関節変数の全成分データ)、設定した値をラジアン単位の数値で格納して処理しするため、信号番号の指定が難しくなります。(モニタなどでは度の単位に変換して表示するので設定値と同じ値を表示します)

例) 出力信号 8 番を指定するため P1.A に 8 を代入 (P1.A=8) しても内部処理は 0.14(rad) を四捨五入して 0 となり、出力信号 0 番を指定することになってしまいます。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

表 4-28 : < 数値 > に記述できる定数・変数

○ : 使用可能、× : 使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度実数 例)M1!	倍精度実数 例)M1#	位置 例)P1.X	関節 例)J1.J1	文字列 例)C1$
M_Out32	○[注 1)]	○[注 1)]	○[注 1)]	○[注 1)]	○	○	○[注 2)]	○	○	×

⚠注意

注 1)

指定された定数が 2 進数でいう 16 ビットに納まる数値 (-32768 ～ +32767) の場合で、ビット 15(16 ビット目) がオンしている数値は負の数値として扱い、上位 16 ビットは符号拡張され全てオン状態の信号を出力しますのでご注意願います。

例) -32768(&B1000000000000000) を指定すると

&B11111111111111111000000000000000(2 進数) が出力されます。

[対策]

下記のように定数を長精度整数型の数値変数に代入した上で、本状態変数 M_YDevD へ代入すると &B00000000000000001000000000000000(2 進数) を出力することができます。

1 M1&=32768

2 M_YDevD(&H20)=M1&

注 2) 出力できる数値の範囲は -2147483648 ～ 2147483647 です。

表 4-29 : < 数値変数 > に記述できる変数

○ : 使用可能、× : 使用不可能

ビット幅	数値変数の型				その他の変数		
	整数 例)M1%	長精度整数 例)M1&	単精度実数 例)M1!	倍精度実数 例)M1#	位置[注 1)] 例)P1.X	関節[注 1)] 例)J1.J1	文字列 例)C1$
M_Out32	×	○	○	○	○	○	×

注 1) 変数で扱う数値が角度の場合 (位置変数の A,B,C 成分データと関節変数の全成分データ)、現在の出力状態をラジアン単位の数値で格納します。モニタなどでは度の単位に変換して表示します。

【関連システム状態変数】

M_Out/M_Outb/M_Out8/M_Outw/M_Out16

【関連命令】

Def IO (デファイオー)

【関連パラメータ】

SYNCIO

M_PI

【機能】
円周率 (3.14159265358979) を返します。

【書式】

例) < 数値変数 >=M_PI

【用語】
< 数値変数 > 代入する数値変数を指定します。

【文例】
1 M1=M_PI 'M1 には 3.14159265358979 が入ります。

【解説】
(1) 代入される変数は実数になります。
(2) 読み込み専用です。

M_Psa

【機能】
指定したタスクスロットがプログラム選択可能かを返します。
1 : プログラム選択可能
0 : 不可 (プログラムが中断状態のとき)

【書式】

例) < 数値変数 >=M_Psa[(< 数式 >)]

【用語】
< 数値変数 > 代入する数値変数を指定します。
< 数式 > 1 ～ 32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】
1 M1=M_Psa(2) 'M1 にはタスクスロット 2 のプログラム選択可状態が入ります。

【解説】
(1) 指定したタスクスロットがプログラム選択可能かを返します。
(2) 読み込み専用です。

M_Ratio

【機能】
ロボットが移動中の目的位置への到達率 (0 ～ 100%) を返します。

【書式】

例) < 数値変数 >=M_Ratio[(< 数式 >)]

【用語】
< 数値変数 > 代入する数値変数を指定します。
< 数式 > 1 ～ 32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】
```
1 Mov P1 WthIf M_Ratio>80, M_Out(1)=1
'P1 へ移動中に目的位置の 80% の距離まで来たら出力信号 1 番を ON します。
```

【解説】
(1) ロボット移動中の特定の位置で処理をおこないたい場合などに利用します。
(2) 読み込み専用です。

M_RCInfo

【機能】
ロボットコントローラの各種情報を返します。

【書式】

例) < 数値変数 >=M_RCInfo(< 情報 ID 番号 >,< 要素番号 >)

【用語】
< 数値変数 > 代入する数値変数を指定します。
< 情報 ID 番号 > 読み出す情報の、情報 ID 番号を指定します。
情報 ID 番号の詳細は、下記の表を参照ください。
< 要素番号 > 読み出す情報に応じて要素番号を指定します。
要素番号の詳細は、下記の表を参照ください。

情報 ID 番号		要素番号	
設定値	情報の内容	設定値 (設定範囲)	意味
1	各軸エンコーダ温度 (℃)	1 ～ 8	ロボットの軸番号を表します。 ※ 存在しない軸および付加軸 (7、8 軸) は、常に 0 を返します。
2	各軸エンコーダ最大温度 (℃)	1 ～ 8	
3	コントローラ盤内温度 (℃)	1	-

【文例】
```
1 Dim METMP(8)                      ' 各軸エンコーダ温度を代入する変数
2 Dim METMPX(8)                     ' 各軸エンコーダ最大温度を代入する変数
3 For MT=1 To 8
4  METMP(MT)=M_RCInfo(1,MT)         ' 各軸エンコーダ温度を読み出す
5  METMPX(MT)=M_RCInfo(2,MT)        ' 各軸エンコーダ最大温度を読み出す
6 Next
7 MRCTMP=M_RCInfo(3,1)              ' コントローラ盤内温度を読み出す
```

【解説】
(1) 情報 ID 番号や要素番号へ設定範囲外の値を指定した場合は、実行時にエラー L3110 が発生します。

M_RDst

【機能】
ロボットが移動中の目的位置までの残距離（mm）を返します。

【書式】

例）<数値変数>=M_RDst[(<数式>)]

【用語】
<数値変数>　　代入する数値変数を指定します。
<数式>　　1～32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】
```
1 Mov P1 WthIf M_RDst<10, M_Out(10)=1    'P1 へ移動中に目的位置残距離が 10mm 以下になったら
                                          出力信号 10 番を ON します。
```

【解説】
(1) ロボット移動中の特定の位置で処理をおこないたい場合などに利用します。
(2) 読み込み専用です。

M_Run

【機能】
指定したタスクスロットのプログラムが実行中かの情報を返します。
1：実行中
0：実行中以外（中断中か停止中）

【書式】

例）<数値変数>=M_Run[(<数式>)]

【用語】
<数値変数>　　代入する数値変数を指定します。
<数式>　　1～32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】
```
1 M1=M_Run(2)          'M1 にはスロット 2 の実効状態が入ります。
```

【解説】
(1) 指定したスロットが実行中の場合は 1 が、停止中（または中断中）の場合は 0 が入ります。
(2) プログラムが停止中（実行行が先頭の場合）かを判断するには M_Run と M_Wai を組み合わせてご利用ください。
(3) 読み込み専用です。

M_SetAdl

【機能】

最適加減速制御が有効 (Oadl On) の際に、指定軸の加減速時間割合を設定します。軸毎に独立して設定できますので負荷が高い軸のモータ負荷を低減するようなことも可能です。また、Ovrd 命令や、Spd 命令、Accel 命令のような全軸一律に設定する方法と違い、タクトタイムへの影響を極力少なくすることが可能です。初期値はパラメータ JADL の値となります。

【書式】

例) M_SetAdl(< 軸番号 >)=< 数値変数 >

【用語】

< 軸番号 >	軸番号を指定します。
< 数値変数 >	標準の加減速時間に対する比率を指定します、1 ～ 100。単位は % です。 初期値は最適加減速補正率ﾊﾟﾗﾒｰﾀ(JADL)値

【文例】

```
1 Accel 100,50                ' 全体の減速割合を 50% に設定します。
2 If M_LdFact(2)>90 Then      'J2 軸の負荷率が 90% を越えたら、
3     M_SetAdl(2)=70          'J2 軸の加減速時間割合を 70% に設定します。
4 EndIf                       ' 加速 70%(=100%×70%)、減速 35%(=50%×70%)
5 Mov P1
6 Mov P2
7 M_SetAdl(2)=100             'J2 軸の加減速時間割合を 100% に戻します。
8 Mov P3                      ' 加速 100%、減速 50%
9 Accel 100,100               ' 全体の減速割合を 100% に戻します。
10 Mov P4
```

【解説】

(1) 軸単位で最適加減速有効時の加減速時間割合を設定することができます。100% を指定した場合、加減速時間は最も短くなります。
(2) 本状態変数により、過負荷エラーやオーバーヒートエラーが発生する軸の負荷を低減するように加減速時間を設定できます。
(3) 本状態変数の設定は、加速時間と減速時間の両方に設定されます。
(4) Accel 命令と併用する場合、最適加減速で算出された加減速時間に対して、さらに Accel 命令の加減速割合していが適用されます。
(5) Accel 命令では、指定した割合で加減速時間が変化します。本状態変数での指定は、軸毎の独立設定であることと、モータの負荷を考慮した加減速時間が算出されることから、加減速時間の変化は、指定した割合とは若干異なる値となることがあります。

【関連命令】

Accel （アクセル）、Ovrd （オーバーライド）、Spd （スピード）

【関連システム状態変数】

M_LdFact

M_SkipCq

【機能】

最後に実行した Skip コマンドを含んだ行の実行を行った結果を返します。
1：Skip した場合
0：Skip しなかった場合

【書式】

例）＜数値変数＞=M_SkipCq[(＜数式＞)]

【用語】

＜数値変数＞　　代入する数値変数を指定します。
＜数式＞　　1 ～ 32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】

```
1 Mov P1 WthIf M_In(10)=1,Skip      'P1 へ移動開始時に入力信号 10 番が入っていたら Mov 命令を
                                     ｽｷｯﾌﾟ
2 If M_SkipCq=1 Then GoTo *Lskip    ' スキップ実行されていたら *Lskip へジャンプ。
 :
10 *Lskip
```

【解説】

(1) スキップ命令が実行されたかの状態をチェックします。
(2) 読み込み専用です。
(3) M_SkipCq 変数を一度でも参照するとスキップ状態がクリアされます（値がゼロになります）。よって、状態を残したい場合は数値変数へ代入して保存してください。

M_Spd/M_NSpd/M_RSpd

【機能】

直線補間、関節補間時の速度の情報を返します。
M_Spd：現在設定されている速度
M_NSpd：初期値（最適速度制御）
M_RSpd：現在の指令速度

【書式】

例）<数値変数>=M_Spd[(<数式>)]

例）<数値変数>=M_NSpd[(<数式>)]

例）<数値変数>=M_RSpd[(<数式>)]

【用語】

<数値変数>	代入する数値変数を指定します。
<数式>	1～32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】

```
1 M1=M_Spd          ' M1 には現在設定されている速度が入ります。
2 Spd M_NSpd        '速度を最適速度制御のモードに戻します。
```

【解説】

(1) M_RSpd はロボットが動作中の現在の指令速度を返します。サーボ OFF 中はフィードバック速度を返します。この時ロボットが停止していても 0.00 ～ ±0.01 の間で値が変化することがあります。
(2) M_RSpd はマルチタスクのプログラムまたは Wth、WthIf 節にて利用します。
(3) 読み込み専用です。

M_SplPno

【機能】

スプライン補間の実行において、最も新しい通過済みの経路点の番号を返します。
スプライン補間の開始位置が１です。
注）この命令はソフトウェアバージョン Ver.R5 以降(F-Q シリーズ)/Ver.S5 以降(F-D シリーズ)で使用できます。

【書式】

例）<数値変数>=M_SplPno[(< メカ番号 >)]

【用語】

<数値変数>　参照結果を代入する数値変数を指定します。
<メカ番号>　スプライン補間を実行しているメカ番号を設定します。
設定範囲：１～３
省略時：１
存在しないメカの番号を指定した場合は、実行時にエラー L3870（指定したメカ番号が無効な値です）が発生します。

【文例】

```
1 Wait M_00=1                          ' スプライン補間が開始するのを待つ
2 *L1:If M_SplPno < 5 Then GoTo *L1    ' 経路点 5 を通過するまで待つ
3 M1=M_Inw(100)                        ' 経路点 5 を通過後、入力信号 100 ～ 115 の状態を取得
4 *L2:If M_SplPno < 10 Then GoTo *L2   ' 経路点 10 を通過するまで待つ
5 M_01=M1                              ' 経路点 10 を通過後、M1 の内容を M_01 に代入
```

【解説】

(1) スプライン補間の実行において、既に通過した経路点の中で最も新しい経路点の番号を返します。
(2) M_SplPno の値をマルチタスクで参照することにより、スプライン補間の進行に応じて演算処理や信号出力等の処理を実行することができるようになります。
(3) M_SplPno は状況に応じて下表に示す値を返します。

表 4-30：M_SplPno が返す値

状況	M_SplPno が返す値
電源投入直後	0
スプライン補間実行中	1～（スプライン補間の進行に応じた値）
スプライン補間終了後	スプライン補間の終了位置の経路点番号
メインプログラムの End 命令実行直後	0
プログラムリセット操作直後	0
スプライン補間に対応していないメカ	0

(4) M_SplPno は読み込み専用の状態変数です。

M_SplVar

【機能】

スプライン補間の実行において、最も新しい通過済みの経路点に設定されていた数値設定値を返します。書き込みにより、任意の値に変更することもできます。

注）この命令はソフトウェアバージョン Ver.R5 以降(F-Q シリーズ)/Ver.S5 以降(F-D シリーズ)で使用できます。

【書式】

例）<数値変数 1>=M_SplVar [(< メカ番号 >)]

例）M_SplVar [(< メカ番号 >)]=< 数値変数 2>

【用語】

<数値変数 1>	参照結果を代入する数値変数を指定します。
< メカ番号 >	スプライン補間を実行しているメカ番号を設定します。 設定範囲：１～３ 省略時：１ 存在しないメカの番号を指定した場合は、実行時にエラー L3870（指定したメカ番号が無効な値です）が発生します。
<数値変数 2>	M_SplVar に設定する値を指定します。 設定範囲：0 ～ 32767 範囲外の値を設定した場合は、実行時にエラー L2615（M_SplVar の設定値が範囲外です)、もしくはエラー L3110（引数の値が範囲外です）が発生します。

【文例】

```
1 Wait M_00=1                    ' スプライン補間が開始するのを待つ
2 *L1
3 Select M_SplVar
4   Case 1                       ' M_SplVar の値に 1 が設定された経路点を通過
5     M_SplVar=0                 ' M_SplVar の値を 0 にリセット
6     M_Out(100)=1               ' 出力信号 100 を ON
7     Break
8   Default                      ' M_SplVar の値が 1 ではない
9     M_Out(100)=0               ' 出力信号 100 を OFF
10    Break
11 End Select
12 If M_00=1 Then Goto *L1       ' スプライン補間が終了するまで繰り返す
```

【解説】

(1) スプライン補間の実行において、既に通過した経路点の中で最も新しい経路点に設定されていた数値設定値を返します。数値設定はスプラインファイルの経路点データの中に指定します。

(2) M_SplVar の値をマルチタスクで参照することにより、スプライン補間の進行に応じて演算処理や信号出力等の処理を実行することができるようになります。M_SplPno とは違い、異なる経路点であっても同じ値を返すことができるので、例えば【文例】のように同一の出力信号の ON/OFF を繰り返す場合にプログラムを簡略化できます。

(3) M_SplVar が返す値は 0 ～ 32767 の整数値です。なお、スプラインファイルの数値設定に "-1" が設定された経路点は、その経路点を通過しても M_SplVar の値は変化せず、経路点通過時の値を継続します。

(4) M_SplVar は状況に応じて表 4-31 に示す値を返します。
表 4-31 : M_SplVar が返す値

状況	M_SplVar が返す値
電源投入直後	0
スプライン補間実行中	スプライン補間の進行に応じた値 (スプラインファイル内の数値設定値)
"-1" が指定された経路点通過時	そのときの値を継続
スプライン補間終了後	スプライン補間の終了時の値
メインプログラムの End 命令実行直後	0
プログラムリセット操作直後	0
M_SplVar に値を書き込んだ直後	書き込んだ値
スプライン補間に対応していないメカ	0 (書き込み実行後も 0)

(5) M_SplVar への書き込み操作により、0 ～ 32767 の範囲において任意の値を設定することができます。スプライン補間や他の操作・処理により値が変更されるまで、その値を保持します。

M_Svo

【機能】
現在のサーボ電源の状態を返します。
1 : サーボ電源 ON
0 : サーボ電源 OFF

【書式】

例) < 数値変数 >=M_Svo[(< メカ番号 >)]

【用語】
< 数値変数 > 代入する数値変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】
```
1 M1=M_Svo(1)          'M1 には現在のサーボ電源の状態が入ります。
```

【解説】
(1) ロボットのサーボ状態を確認できます。
(2) 読み込み専用です。

M_Timer

【機能】

ミリ秒単位で時間をカウントしています。ロボットの動作時間測定や正確な時間を計測する場合に使用できます。

【書式】

例)<数値変数>=M_Timer(<数式>)

【用語】

<数値変数>　　代入する数値変数を指定します。
<数式>　　1～8の番号を入れます。カッコは省略できません。

【文例】

```
1 M_Timer(1)=0
2 Mov P1
3 Mov P2
4 M1=M_Timer(1)          'M1にはP1からP2までの移動時間(ms)が入ります。
                          例) 5.346秒の場合、M1の値は5346になります。
5 M_Timer(1)=1.5         '1.5に設定します。
```

【解説】

(1) 代入も可能です。代入時の単位は秒で設定されます。
(2) ミリ秒(ms)単位で計測できるため正確なタクト時間測定が可能です。

M_Tool

【機能】
指定した番号のツールデータ（MEXTL1 ～ 16）を現在のツールデータとして使用すると共に、パラメータ MEXTL に設定します。
また現在のツール番号の読み出しも可能です。

【書式】

例）＜数値変数＞=M_Tool[(＜メカ番号＞)]　　’現在のツール番号の参照
例）M_Tool[(＜メカ番号＞)]=＜数式＞　　’ツール番号を設定

【用語】

＜数値変数＞	代入する数値変数を指定します。
＜メカ番号＞	1 ～ 3 のメカ番号を入れます。省略時は 1 です。
＜数式＞	1 ～ 16 のツール番号を入れます。

【文例】

①ツールデータの設定

```
1 Tool (0,0,100,0,0,0)          ’ツールデータ (0,0,100,0,0,0) を指定し、MEXTL に書き込みます。
2 Mov P1
3 M_Tool=2                      ’ツールデータをツール番号 2(MEXTL2) の値に変更します。
4 Mov P2
```

②ツール番号の参照

```
1 If M_In(900)=1 Then           ’ハンド入力信号によりツールデータを切り替える。
2   M_Tool=1                    ’ ツール 1 をツールデータへ設定します。
3 Else
4   M_Tool=2                    ’ツール 2 をツールデータへ設定します。
5 EndIf
6 Mov P1
```

【解説】

(1) ツール用パラメータ MEXTL1 ～ 16 に設定されている値をツールデータに反映します、パラメータ MEXTL にも書き込まれます。
(2) ツール番号 1 ～ 16 は MEXTL1 ～ 16 に対応します。
(3) 参照時は現在設定されているツール番号を読み出します。
(4) 読み出した値が 0 の時は、MEXTL1 ～ 16 以外のツールデータが現在のツールデータとして設定されています。
(5) T/B のツール設定画面でも同様の設定が可能です。詳細は 28 ページの「3.2.9 ツールデータの切替」を参照してください。

【関連パラメータ】
MEXTL、MEXTL1 ～ 16

【関連命令】
Tool（ツール）

M_Uar

【機能】

ユーザ定義領域に入っているかを返します。
ビット 0 ～ 15 が領域 1 ～ 16 に対応し、それぞれのビットが下記の情報を表示します。
1：ユーザ定義領域内、
0：ユーザ定義領域外

【書式】

例）<数値変数>=M_Uar[(< メカ番号 >)]

【用語】

<数値変数>	代入する数値変数を指定します。
<メカ番号>	メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 M1 = M_Uar(1) AND &H0004    'M1 に領域 3 のみの結果を代入。
2 If M1<>0 Then M_Out(10)=1   ' 領域 3 に入っていれば出力信号 10 を ON します。
```

【解説】

(1) ユーザ定義領域の使用方法は、435 ページの「5.8 ユーザ定義領域について」を参照願います。
(2) 領域 1 ～ 16 が対象となります。領域 17 ～ 32 の情報が必要な場合には M_Uar32 を使用して取得してください。
(3) パラメータ AREAnAT(n は領域番号 (n=1 ～ 16)) に 1(信号出力) を指定した領域が対象となります。
(4) 比較演算、論理演算を行なう場合には、10 進数表記ではビット 15 が 1 の場合に負の数となりますので、16 進表記を使用されることを推奨します。
(5) 読込専用です。

【関連システム変数】

M_Uar32

【M_Uar の値とユーザ定義領域の対応】

ビット	領域	10 進数	16 進数
0	1	1	&H0001
1	2	2	&H0002
2	3	4	&H0004
3	4	8	&H0008
4	5	16	&H0010
5	6	32	&H0020
6	7	64	&H0040
7	8	128	&H0080

ビット	領域	10 進数	16 進数
8	9	256	&H0100
9	10	512	&H0200
10	11	1024	&H0400
11	12	2048	&H0800
12	13	4096	&H1000
13	14	8192	&H2000
14	15	16384	&H4000
15	16	-32768	&H8000

例）ユーザ定義領域 5 及び 10 に入っている場合には、領域 5 を示す値 &H0010 と、領域 10 を示す値 &H0200 を足した値である &H0410 が、M_Uar の値として返されます。

M_Uar32

【機能】

ユーザ定義領域に入っているかを返します。
ビット 0 ～ 31 が領域 1 ～ 32 に対応し、それぞれのビットが下記の情報を表示します。
1：ユーザ定義領域内
0：ユーザ定義領域外

【書式】

例）＜数値変数＞=M_Uar32[(＜メカ番号＞)]

【用語】

＜数値変数＞　　代入する数値変数を指定します。
＜メカ番号＞　　メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 Def Long M1
2 M1& = M_Uar32(1) AND &H00080000        'M1 に領域 20 のみの結果を代入。
3 If M1&<>0 Then M_Out(10)=1             ' 領域 20 に入っていれば出力信号 10 を ON します。
```

【解説】

(1) ユーザ定義領域の使用方法は、435 ページの「5.8 ユーザ定義領域について」を参照願います。
(2) ＜数値変数＞に 16 ビット整数型を使用し、値がオーバーした場合はエラーとなります。
32 ビット整数型を使用してください。
(3) パラメータ AREAnAT(n は領域番号 (n=1 ～ 32)) に 1(信号出力) を指定した領域が対象となります。
(4) 比較演算、論理演算を行なう場合には、10 進数表記ではビット 31 が 1 の場合に負の数となりますので、16 進数表記を使用されることを推奨します。
(5) 読込専用です。

【関連システム変数】

M_Uar

【M_Uar32 の値とユーザ定義領域の対応】

ビット	領域	10 進数	16 進数
0	1	1	&H00000001
1	2	2	&H00000002
2	3	4	&H00000004
3	4	8	&H00000008
4	5	16	&H00000010
5	6	32	&H00000020
6	7	64	&H00000040
7	8	128	&H00000080
8	9	256	&H00000100
9	10	512	&H00000200
10	11	1024	&H00000400
11	12	2048	&H00000800
12	13	4096	&H00001000
13	14	8192	&H00002000
14	15	16384	&H00004000
15	16	32768	&H00008000

ビット	領域	10 進数	16 進数
16	17	65536	&H00010000
17	18	131072	&H00020000
18	19	262144	&H00040000
19	20	524288	&H00080000
20	21	1048576	&H00100000
21	22	2097152	&H00200000
22	23	4194304	&H00400000
23	24	8388608	&H00800000
24	25	16777216	&H01000000
25	26	33554432	&H02000000
26	27	67108864	&H04000000
27	28	134217728	&H08000000
28	29	268435456	&H10000000
29	30	536870912	&H20000000
30	31	1073741824	&H40000000
31	32	-2147483648	&H80000000

例）ユーザ定義領域 5 及び 10 に入っている場合には、領域 5 を示す値 &H00000010 と、領域 10 を示す値 &H00000200 を足した値である &H00000210 が、M_Uar の値として返されます。

M_UDevW／M_UDevD

【機能】

CR7xx-Q シリーズのコントローラでマルチ CPU 間共有メモリへの直接書き込み、または直接参照をおこないます。(CR7xx-Q シリーズコントローラのみ)
(シーケンサのラダープログラムを介さずに、複数のロボット間でより高速にインターロックなどのインタフェースを取ることができます。また、モーション CPU などロボット CPU 以外の共有メモリ情報の取込みも可能です。(506 ページの「5.26 ロボット CPU 間直接通信」を参照))

M_UDevW : ワード単位 (16 ビット分) で書き込み / 参照します。
M_UDevD : ダブルワード単位 (32 ビット分) で書き込み / 参照します。

【書式】

参照
< 数値変数 >=M_UDevW(< 先頭入出力番号 >,< 共有メモリアドレス >)
< 数値変数 >=M_UDevD(< 先頭入出力番号 >,< 共有メモリアドレス >)

書込み
M_UDevW(< 先頭入出力番号 >,< 共有メモリアドレス >)=< 数値 >
M_UDevD(< 先頭入出力番号 >,< 共有メモリアドレス >)=< 数値 >

【用語】

< 数値変数 > 入出力信号の値を代入する数値変数を指定します。

< 先頭入出力番号 > CPU ユニットの入出力番号を定数または数値変数で指定します。
(指定する値は先頭入出力番号を 16 進表現し、下位 1 桁を省略した値です)
範囲 : 16 進表現で &H3E0 ～ &H3E3(10 進表現で 992 ～ 995)
1 号機 : &H3E0(10 進表現では 992)
2 号機 : &H3E1(10 進表現では 993)
3 号機 : &H3E2(10 進表現では 994)
4 号機 : &H3E3(10 進表現では 995)
注) データの書き込みは自号機のみが有効です。

< 共有メモリアドレス > CPU ユニットの共有メモリアドレスを定数または数値変数で指定します。
有効範囲は各状態変数毎に以下となります。(10 進数)
M_UDevW : 10000 ～ 24335
M_UDevD : 10000 ～ 24334

< 数値 > 書込むデータを定数または数値変数で指定します。
有効範囲は各状態変数毎に以下となります。
M_UDevW : -32768 ～ 32767(&H8000 ～ &H7FFF)
M_UDevD : -2147483648 ～ 2147483647(&H80000000 ～ &H7FFFFFFF)

【文例】

```
1 M_UDevW(&H3E1, 10010)=&HFFFF        ' 2 号機 CPU( 自号機 ) の共有メモリアドレス 10010 に &HFFFF
                                        (16 進数 ) を書き込む
2 M_UDevD(&H3E1, 10011)=P1.X * 1000   ' 2 号機 CPU( 自号機 ) の共有メモリアドレス 10011/10012 の 2
                                        ワードに位置変数 P1 の X 座標値を 1000 倍した値を書き込む
3 M1%=M_UDevW(&H3E2, 10001) And &H7   ' M1 に 3 号機 CPU の共有メモリアドレス 10001 の下位 3 ビッ
                                        ト分の値を代入する
```

【解説】

(1) シーケンサのマルチ CPU 間共有メモリへの書き込み、または参照をします。
(2) 対象の共有メモリは先頭入出力番号と共有メモリアドレスで指定します。
(3) 書込むデータ、または参照で返すデータはいずれも整数値です。
(4) 指定した共有メモリアドレスから M_UDevW は 1 ワード (16 ビット)、M_UDevD は 2 ワード (32 ビット) 分のデータをそれぞれ対象とします。

(5) 先頭入出力番号は 16 進表現で &H3E0 ～ &H3E3(10 進表現で 992 ～ 995)、共有メモリアドレスは書き込みまたは参照するアドレスが 10 進表現で 10000 ～ 24335 になるように指定ください。
(6) 書き込みは自号機 CPU の共有メモリアドレスのみ可能です。他号機 CPU のアドレスを指定してデータを書込んでも更新されません。
(7) M_UDevD で 32 ビットデータの泣き別れを防止するためには、共有メモリの偶数アドレスを先頭にしてデータへアクセスしてください。

【補足】

表 4-32 : < 数値変数 > に記述できる変数

○ : 使用可能、× : 使用不可能

ビット幅	数値変数の型				その他の変数		
	整数 例)M1%	長精度整数 例)M1&	単精度 実数 例)M1!	倍精度 実数 例)M1#	位置注 1) 例)P1.X	関節注 1) 例)J1.J1	文字列 例)C1$
M_UDevW	×	○	○	○	○	○	×
M_UDevD	×	○	○	○	○	○	×

注 1) 変数で扱う数値が角度の場合 (位置変数の A,B,C 成分データと関節変数の全成分データ)、現在の出力状態をラジアン単位の数値で格納し演算などの処理はラジアン単位で実行します。モニタなどでは度の単位に変換して表示します。

表 4-33 : < シーケンサ入力信号番号 >< 共有メモリアドレス > に記述できる定数・変数

ビット幅	定数の型			数値変数の型				その他の変数		
	数値注 1) 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度 実数注 1) 例)M1!	倍精度 実数注 1) 例)M1#	位置 注 1) ,注 2) 例)P1.X	関節 注 1) , 注 2) 例)J1.J1	文字列 例)C1$
M_UDevW	○	○	○	○	○	○	○	○	×	×
M_UDevD	○	○	○	○	○	○	○	○	×	×

○ : 使用可能、× : 使用不可能

注 1) 実数値は四捨五入されます。
注 2) 変数で扱う数値が角度の場合 (位置変数の成分データ A,B,C と関節変数の全成分データ)、設定した値をラジアン単位の数値で格納して処理しするため、信号番号の指定が難しくなります。(モニタなどでは度の単位に変換して表示するので設定値と同じ値を表示します)
例) 出力信号 8 番を指定するため P1.A に 8 を代入 (P1.A=8) しても内部処理は 0.14(rad) を四捨五入して 0 となり、出力信号 0 番を指定することになってしまいます。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

表 4-34 : < 数値 > に記述できる定数・変数

○ : 使用可能、× : 使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値 注 1) 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度 実数 例)M1!	倍精度 実数 例)M1#	位置 例)P1.X	関節 例)J1.J1	文字列 例)C1$
M_UDevW	○	○	○	○	×	×	×	×	×	×
M_UDevD	○注 2)	○注 2)	○注 2)	○注 2)	○	○	○注 3)	○	○	×

注 1) 実数値は四捨五入されます。

注 2) ⚠注意 指定された定数が 2 進数でいう 16 ビットに納まる数値 (-32768 ～ +32767) の場合で、ビット 15(16 ビット目) がオンしている数値は負の数値として扱い、上位 16 ビットは符号拡張され全てオン状態の信号を出力しますのでご注意願います。

例) -32768(&B1000000000000000) を指定すると
&B11111111111111111000000000000000(2 進数) を出力します

[対策]
下記のように定数を長精度整数型の数値変数に代入した上で、本状態変数 M_YDevD へ代入すると &B00000000000000001000000000000000(2 進数) を出力することができます。

```
1 M1&=32768
2 M_YDevD(&H20)=M1&
```

注 3) 出力できる数値の範囲は -2147483648 ～ 2147483647 です。

【参考】号機間データのデータ保証について

自号機の読み出しと、他号機のデータ書き込み / 他号機からの受信タイミングにより、号機ごとのデータに古いデータと新しいデータが混在（データの泣き別れ）することがあります。
CPU 共有メモリを使用したプログラムによる交信において、データの泣き別れを防止する方法を図 4-34 に示します。

1) 32 ビットデータの泣き別れ防止

CPU 共有メモリのユーザー自由エリアへは、偶数アドレス（例：アドレス 10002）を先頭にしてアクセスすることにより 32 ビットデータの泣き別れを防止することができます。

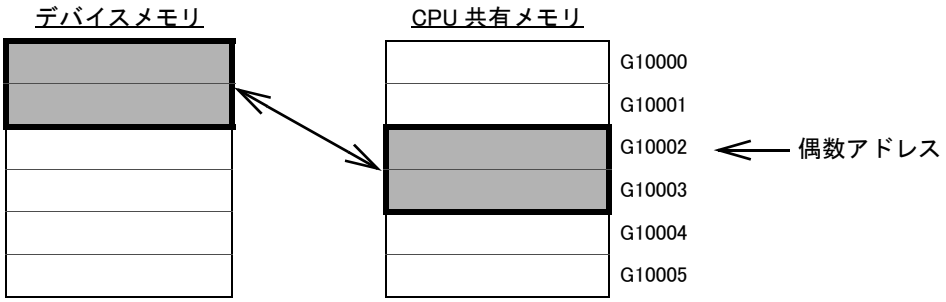


図 4-34：32 ビットデータの泣き別れ防止

2) 32 ビットを超えるデータの泣き別れ防止

プログラムによる読み出しは、ユーザー自由エリアの先頭から順に読み出します。
また書き込み命令ではユーザー自由エリアの最終アドレスから先頭アドレスに向かって、送信データを書き込みます。
そのため交信を行うデータの先頭にインタロック用のデバイスを設けることにより、交信を行うデータの泣き別れを防止することができます。

M_Wai

【機能】
指定したタスクスロットのプログラムの待機状態を返します。
1：中断中（プログラムが中断状態）
0：中断中以外（運転中か停止中）

【書式】

例）<数値変数>=M_Wai[(<数式>)]

【用語】
<数値変数>　　代入する数値変数を指定します。
<数式>　　1～32、タスクスロット番号を指定します。省略時は現在のスロットです。

【文例】
```
1 M1=M_Wai(1)          'M1 にはスロット 1 の待機状態が入ります。
```

【解説】
(1) プログラムが中断状態かをチェックするのに使用します。
(2) プログラムが停止中（実行行が先頭の場合）かを判断するには M_Run と M_Wai を組み合わせてご利用ください。
(3) 読み込み専用です。

【関連システム状態変数】
M_Wupov、M_Wuprt、M_Wupst

M_Wupov

【機能】

暖機運転状態のとき動作速度を下げるために指令速度に掛けるオーバーライド（暖機運転オーバーライド、単位：%）の値を返します。

注）暖機運転モードの詳細については461 ページの「5.19 暖機運転モードについて」を参照ください。

【書式】

例）<数値変数>=M_Wupov[(<メカ番号>)]

【用語】

<数値変数>	代入する数値変数を指定します。
<メカ番号>	メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 M1=M_Wupov(1)    ' M1 に暖機運転オーバーライドの値が入ります。
```

【解説】

(1) ロボットが暖機運転状態（速度を自動的に下げて動作する状態）となっているとき、動作速度を下げるために指令速度に掛けるオーバーライド（暖機運転オーバーライド）の値を確認するときに使用します。

(2) 暖機運転モードが無効の場合、およびコントローラのモードが"MANUAL"となっている場合、マシンロック中は、常に 100 となります。

(3) 通常の状態から暖機運転状態へ変化したとき、あるいは電源投入直後に暖機運転状態であった場合、パラメータ WUPOVRD の第 1 要素（暖機運転オーバーライドの初期値）で指定した値が初期値として設定され、ロボットの動作に応じて M_Wupov の値は大きくなっていきます。そして暖機運転状態が解除されたとき、M_Wupov の値は 100 となります。

(4) 暖機運転状態における実際のオーバーライドは、以下のとおりです。

・関節補間動作時 =（操作パネル(T/B)のオーバーライド設定値）×（プログラムオーバーライド(Ovrd 命令)）×（関節オーバーライド(JOvrd 命令)）× 暖機運転オーバーライド

・直線補間動作時 =（操作パネル(T/B)のオーバーライド設定値）×（プログラムオーバーライド(Ovrd 命令)）×（直線指定速度(Spd 命令)）× 暖機運転オーバーライド

(5) 読み込み専用です。

M_Wuprt

【機能】

暖機運転状態を解除するために対象軸が動作しなければならない時間（秒）を返します。
注）暖機運転モードの詳細については461 ページの「5.19 暖機運転モードについて」を参照ください。

【書式】

例）＜数値変数＞=M_Wuprt[(＜メカ番号＞)]

【用語】

＜数値変数＞　　代入する数値変数を指定します。
＜メカ番号＞　　メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
10 M1=M_Wuprt(1)  ' M1 に対象軸が動作しなければならない時間が入ります。
```

【解説】

(1) ロボットが暖機運転状態（速度を自動的に下げて動作する状態）となっているとき、パラメータ WUPAXIS（暖機運転モード対象軸）で指定した関節軸があとどのくらいの時間動作すれば、暖機運転状態が解除されるかを確認するときに使用します。
(2) 暖機運転モードが無効の場合は、常に 0 を返します。
(3) 通常の状態から暖機運転状態へ変化したとき、あるいは電源投入直後に暖機運転状態であった場合、パラメータ WUPTIME の第 1 要素（暖機運転モードの有効時間）で指定した時間が初期値として設定され、ロボットの動作に応じて M_Wuprt の値は小さくなっていきます。そして値が 0 となったときに暖機運転状態が解除されます。
(4) 暖機運転モードの対象軸が複数存在する場合は、その中で最も動作時間の短い軸の値を返します。
例えば、ある対象軸(A)が動作して残り 20 秒で暖機運転状態が解除されるとき(M_Wuprt=20 のとき)に、停止し続けていた別の対象軸(B)が通常の状態から暖機運転状態へ変化すると、(B)が最も動作時間の短い軸（動作時間は 0 秒）となるので、(B)の動作しなければならない時間(= 暖機運転モードの有効時間、初期値は 60 秒）がこの状態変数の値(M_Wuprt=60)となります。
(5) 読み込み専用です。

M_Wupst

【機能】

暖機運転状態が解除された後、再び暖機運転状態となるまでの時間（秒）を返します。
注）暖機運転モードの詳細については461 ページの「5.19 暖機運転モードについて」を参照ください。

【書式】

例）<数値変数>=M_Wupst[(<メカ番号>)]

【用語】

<数値変数>　　代入する数値変数を指定します。
<メカ番号>　　メカ番号を入れます。1～3、省略時は 1 です。

【文例】

```
1 M1=M_Wupst(1)    ' M1 に再び暖機運転状態となるまでの時間が入ります。
```

【解説】

(1) ロボットの暖機運転状態（速度を自動的に下げて動作する状態）が解除されているとき、パラメータ WUPAXIS（暖機運転モード対象軸）で指定した関節軸があとどのくらいの時間停止し続けると暖機運転状態となるかを確認するときに使用します。
(2) 暖機運転モードが無効の場合は、パラメータ WUPTIME の第 2 要素（暖機運転モードの再開時間）に指定した時間を返します。
(3) 暖機運転状態が解除されているときに対象軸が動作すると、パラメータ WUPTIME の第 2 要素（暖機運転モードの再開時間）に指定した時間が初期値として設定され、対象軸が停止していると M_Wupst の値は小さくなっていきます。そして値が 0 となったときに暖機運転状態となります。
(4) 対象軸が複数存在する場合は、その中で最も停止し続けている軸の値を返します。
(5) 読み込み専用です。

M_XDev/M_XDevB/M_XDevW/M_XDevD

【機能】

CR7xx-Q シリーズのコントローラでシーケンサ入力信号 (X) の値を参照します。(CR7xx-Q シリーズコントローラのみ)
(他号機が管理する入出力ユニット / 入出力混合ユニットの入力信号を直接参照します。(502 ページの「5.25 シーケンサ入出力ユニット直接制御」を参照))

M_XDev : ビット単位で参照します。
M_XDevB : バイト単位 (8 ビット分) で参照します。
M_XDevW : ワード単位 (16 ビット分) で参照します。
M_XDevD : ダブルワード単位 (32 ビット分) で参照します。

【書式】

```
< 数値変数 >=M_XDev[(< シーケンサ入力信号番号 >)]
< 数値変数 >=M_XDevB[(< シーケンサ入力信号番号 >)]
< 数値変数 >=M_XDevW[(< シーケンサ入力信号番号 >)]
< 数値変数 >=M_XDevD[(< シーケンサ入力信号番号 >)]
```

【用語】

< 数値変数 >	入力信号の値を代入する数値変数を指定します。
< シーケンサ入力信号番号 >	参照する入力信号番号を定数または数値変数で指定します。 設定範囲は 16 進表現で &H0 ～ &HFFF(10 進表現で 0 ～ 4095) ですが、各状態変数毎に以下となります。 M_XDev : &H0 ～ &HFFF(0 ～ 4095) M_XDevB : &H0 ～ &HFF8(0 ～ 4088) M_XDevW : &H0 ～ &HFF0(0 ～ 4080) M_XDevD : &H0 ～ &HFE0(0 ～ 4064) 注) 実数の数値変数で指定した場合は四捨五入して設定します。

【文例】

```
1 M1%=M_XDev(1)                 ' M1 にシーケンサ入力信号 1 番の値 (1 か 0) を代入
2 M2%=M_XDevB(&H10)             ' M2 にシーケンサ入力信号 10 番 (16 進数 ) から 8 ビット分の値を代入
3 M3%=M_XDevW(&H20) And &H7     ' M3 にシーケンサ入力信号 20 番 (16 進数 ) から 3 ビット分の値を代入
4 M4%=M_XDevW(&H20)             ' M4 にシーケンサ入力信号 20 番 (16 進数 ) から 16 ビット分の値を
                                  代入
5 M5&=M_XDevD(&H100)            ' M5 にシーケンサ入力信号 100 番 (16 進数 ) から 32 ビット分の値を
                                  代入
6 P1.Y=M_XDevD(&H100)/1000      ' 位置変数 P1 の Y 成分にシーケンサ入力信号 100 番 (16 進数 ) から 32
                                  ビット分のデータを整数値として入力し、1/1000 倍して ( 実数化 )
                                  代入
```

【解説】

(1) シーケンサ入力信号 (X) の状態を整数値で返します。
(2) 指定したシーケンサ入力信号番号から M_XDev は 1 ビット、M_XDevB は 8 ビット、M_XDevW は 16 ビット、M_XDevD は 32 ビット分のデータをそれぞれ返します。
(3) シーケンサ入力信号番号は、参照する信号範囲が 16 進表現で &H0 ～ &HFFF(10 進表現で 0 ～ 4095) になるように指定ください。省略や、範囲外の場合は、L3110(引数の値が範囲外) が発生します。
(4) 予めパラメータ QXYREAD で入力信号の参照を有効に設定しておく必要があります。
(5) 対応するシーケンサユニットが接続されていない時は 0 を返します。

【補足】

表 4-35：＜数値変数＞に記述できる変数

○：使用可能、×：使用不可能

ビット幅	数値変数の型				その他の変数		
	整数 例)M1%	長精度整数 例)M1&	単精度 実数 例)M1!	倍精度 実数 例)M1#	位置注 1) 例)P1.X	関節注 1) 例)J1.J1	文字列 例)C1$
M_XDev	○	○	○	○	○	○	×
M_XDevB	○	○	○	○	○	○	×
M_XDevW	○	○	○	○	○	○	×
M_XDevD	×	○	○	○	○	○	×

注 1）変数で扱う数値が角度の場合（位置変数の A,B,C 成分データと関節変数の全成分データ）、現在の出力状態をラジアン単位の数値で格納し演算などの処理はラジアン単位で実行します。モニタなどでは度の単位に変換して表示します。

表 4-36：＜シーケンサ入力信号番号＞に記述できる定数・変数

○：使用可能、×：使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値注 1) 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度 実数注 1) 例)M1!	倍精度 実数注 1) 例)M1#	位置 注 1），注 2) 例)P1.X	関節 注 1），注 2) 例)J1.J1	文字列 例)C1$
M_XDev	○	○	○	○	○	○	○	○	×	×
M_XDevB	○	○	○	○	○	○	○	○	×	×
M_XDevW	○	○	○	○	○	○	○	○	×	×
M_XDevD	○	○	○	○	○	○	○	○	×	×

注 1）実数値は四捨五入されます。

注 2）変数で扱う数値が角度の場合（位置変数の成分データ A,B,C と関節変数の全成分データ）、設定した値をラジアン単位の数値で格納して処理するため、信号番号の指定が難しくなります。（モニタなどでは度の単位に変換して表示するので設定値と同じ値を表示します）

例）出力信号 8 番を指定するため P1.A に 8 を代入 (P1.A=8) しても内部処理は 0.14(rad) を四捨五入して 0 となり、出力信号 0 番を指定することになってしまいます。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

M_YDev/M_YDevB/M_YDevW/M_YDevD

【機能】

CR7xx-Q シリーズのコントローラでシーケンサ出力信号 (Y) への書き込み、または参照します。(CR7xx-Q シリーズコントローラのみ)
(指定した入出力ユニット / 入出力混合ユニットをロボット CPU が管理するように設定しておき、出力信号の直接参照、または直接書き込をおこないます。(502 ページの「5.25 シーケンサ入出力ユニット直接制御」を参照))

M_YDev ：ビット単位で書き込み / 参照します。
M_YDevB：バイト単位 (8 ビット分) で書き込み / 参照します。
M_YDevW：ワード単位 (16 ビット分) で書き込み / 参照します。
M_YDevD：ダブルワード単位 (32 ビット分) で書き込み / 参照します。

【書式】

参照
< 数値変数 >=M_YDev[(< シーケンサ出力信号番号 >)]
< 数値変数 >=M_YDevB[(< シーケンサ出力信号番号 >)]
< 数値変数 >=M_YDevW[(< シーケンサ出力信号番号 >)]
< 数値変数 >=M_YDevD[(< シーケンサ出力信号番号 >)]

書込み
M_YDev[(< シーケンサ出力信号番号 >)]=< 数値 >
M_YDevB[(< シーケンサ出力信号番号 >)]=< 数値 >
M_YDevW[(< シーケンサ出力信号番号 >)]=< 数値 >
M_YDevD[(< シーケンサ出力信号番号 >)]=< 数値 >

【用語】

< 数値変数 > 参照する出力信号の値を代入する数値変数を指定します。

< シーケンサ入力信号番号 > 参照、または書き込む出力信号番号を定数または数値変数で指定します。
設定範囲は 16 進表現で &H0 ～ &HFFF(10 進表現で 0 ～ 4095) ですが、各状態変数により以下となります。
M_YDev ：&H0 ～ &HFFF(0 ～ 4095)
M_YDevB：&H0 ～ &HFF8(0 ～ 4088)
M_YDevW：&H0 ～ &HFF0(0 ～ 4080)
M_YDevD：&H0 ～ &HFE0(0 ～ 4064)
注) 実数の数値変数で指定した場合は四捨五入して設定します。

< 数値 > 出力するデータを定数、または変数で指定します。
有効範囲は各状態変数毎に以下となります。
M_YDev ：0 または 1(&H0 ～ &H1)
M_YDevB：-128 ～ 127(&H80 ～ &H7F)
M_YDevW：-32768 ～ 32767(&H8000 ～ &H7FFF)
M_YDevD：-2147483648 ～ 2147483647(&H80000000 ～ &H7FFFFFFF)
注) 実数で指定した場合は四捨五入して設定します。

【文例】

```
1 M_YDev(1)=1                ' シーケンサ出力信号 1 番を ON します
2 M_YDevB(&H10)=&HFF         ' シーケンサ出力信号 10 番 (16 進数 ) から 8 ビット分を ON します
3 M_YDevW(&H20)=&HFFFF       ' シーケンサ出力信号 20 番 (16 進数 ) から 16 ビット分を ON します
4 M_YDevD(&H100)=P1.X * 1000 ' シーケンサ出力信号 100 番 (16 進数 ) から 32 ビット分に位置変数 P1
                               の X 座標値 1000 倍して格納します
5 M1%=M_YDevW(&H20) And &H7  ' M1 にシーケンサ出力信号 20 番 (16 進数 ) から 3 ビット分の値を代入
```

【解説】

(1) シーケンサ出力信号 (Y) へ書き込み、または参照をします。
(2) 書込むデータ、または参照して返すデータはいずれも整数値です。

(3) 指定したシーケンサ出力信号番号から M_YDev は 1 ビット、M_YDevB は 8 ビット、M_YDevW は 16 ビット、M_YDevD は 32 ビット分のデータをそれぞれ対象とします。
(4) シーケンサ出力信号番号は、書き込みまたは参照する信号範囲が 16 進表現で &H0 ～ &HFFF(10 進表現で 0 ～ 4095) になるように指定ください。省略や、範囲外の場合は、L3110(引数の値が範囲外) が発生します。
(5) 予めパラメータ QXYREAD で入力信号の参照を有効に、またパラメータ QXYUNITn(n=1 ～ 4) でロボット CPU が管理する入出力ユニットを設定しておく必要があります。
(6) Dly 命令を組合わせてパルス出力は対応していません。Dly 命令を使用すると、L4220(入力した命令文の構文に誤りがあります。) が発生します。
(7) 書き込み時に対応するシーケンサユニットが接続されていない時、信号は変化しません。
参照時に対応するシーケンサユニットが接続されていない時は 0 を返します。

【補足】

表 4-37 : < 数値変数 > に記述できる変数　　○ : 使用可能、× : 使用不可能

ビット幅	数値変数の型				その他の変数		
	整数 例)M1%	長精度整数 例)M1&	単精度 実数 例)M1!	倍精度 実数 例)M1#	位置注 1) 例)P1.X	関節注 1) 例)J1.J1	文字列 例)C1$
M_YDev	○	○	○	○	○	○	×
M_YDevB	○	○	○	○	○	○	×
M_YDevW	○	○	○	○	○	○	×
M_YDevD	×	○	○	○	○	○	×

注 1) 変数で扱う数値が角度の場合 (位置変数の A,B,C 成分データと関節変数の全成分データ)、現在の出力状態をラジアン単位の数値で格納し演算などの処理はラジアン単位で実行します。モニタなどでは度の単位に変換して表示します。

表 4-38 : < シーケンサ入力信号番号 > に記述できる定数・変数　　○ : 使用可能、× : 使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値注 1) 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度 実数注 1) 例)M1!	倍精度 実数注 1) 例)M1#	位置 注 1) ,注 2) 例)P1.X	関節 注 1) , 注 2) 例)J1.J1	文字列 例)C1$
M_YDev	○	○	○	○	○	○	○	○	×	×
M_YDevB	○	○	○	○	○	○	○	○	×	×
M_YDevW	○	○	○	○	○	○	○	○	×	×
M_YDevD	○	○	○	○	○	○	○	○	×	×

注 1) 実数値は四捨五入されます。
注 2) 変数で扱う数値が角度の場合 (位置変数の成分データ A,B,C と関節変数の全成分データ)、設定した値をラジアン単位の数値で格納して処理するため、信号番号の指定が難しくなります。(モニタなどでは度の単位に変換して表示するので設定値と同じ値を表示します)
例) 出力信号 8 番を指定するため P1.A に 8 を代入 (P1.A=8) しても内部処理は 0.14(rad) を四捨五入して 0 となり、出力信号 0 番を指定することになってしまいます。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

表 4-39：＜数値＞に記述できる定数・変数

○：使用可能、×：使用不可能

ビット幅	定数の型			数値変数の型				その他の変数		
	数値 注 1) 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度 実数 例)M1!	倍精度 実数 例)M1#	位置 例)P1.X	関節 例)J1.J1	文字列 例)C1$
M_YDev	○	○	○	○	×	×	×	×	×	×
M_YDevB	○	○	○	○	×	×	×	×	×	×
M_YDevW	○	○	○	○	×	×	×	×	×	×
M_YDevD	○注 2)	○注 2)	○注 2)	○注 2)	○	○	○注 3)	○	○	×

注 1）実数値は四捨五入されます。

注 2）⚠注意　指定された定数が 2 進数でいう 16 ビットに納まる数値(-32768 ～ +32767)の場合で、ビット 15(16 ビット目)がオンしている数値は負の数値として扱い、上位 16 ビットは符号拡張され全てオン状態の信号を出力しますのでご注意願います。

例) -32768(&B1000000000000000)を指定すると
&B11111111111111111000000000000000(2 進数)を出力します。

[対策]
下記のように定数を長精度整数型の数値変数に代入した上で、本状態変数 M_YDevD へ代入すると &B00000000000000001000000000000000(2 進数)を出力することができます。

1 M1&=32768
2 M_YDevD(&H20)=M1&

注 3）出力できる数値の範囲は -2147483648 ～ 2147483647 です。

P_Base/P_NBase

【機能】

ベース変換データに関する情報を返します。
P_Base : 現在設定されている BASE 変換データを返します。
P_NBase : 初期値 (0,0,0,0,0,0)(0,0) を返します。

【書式】

例) < 位置変数 >=P_Base[(< メカ番号 >)]

例) < 位置変数 >=P_NBase

【用語】

< 位置変数 > 代入する位置変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 P1=P_Base          'P1 には現在設定されているベース変換データが入ります。
2 Base P_NBase       ' ベース変換データを初期値に戻します。
```

【解説】

(1) P_NBase には (0,0,0,0,0,0)(0,0) が入っています。
(2) ベース変換はティーチングデータに影響がでますので慎重にご使用ください。
(3) ベース位置を変更する場合は Base 命令をご使用ください。
(4) 読み込み専用です。

P_CavDir

【機能】

干渉チェックで干渉を検出した時のロボットの動作方向を返します。
本機能は対応機種を限定しています。詳細については「5.24 干渉回避機能について」を参照してください。

【書式】

例）<位置変数>=P_CavDir[(<メカ番号>)]

【用語】

<位置変数>　　代入する位置変数を指定します。
<メカ番号>　　メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

500 ページの「5.24.10 サンプルプログラム」を参照。

【解説】

(1) 干渉チェック後の自動復旧動作でロボットの動作方向を確認する際に使用します。
(2) 干渉を検知した瞬間のロボットの動作方向を、最大移動軸の値を ±1.0 とした比率で表します。
　例）(X 軸方向：Y 軸方向)=(2：−1) の比率で動作していた場合 .....P_CavDir=(1,−0.5,0,0,0,0)(0,0)
(3) 姿勢軸と構造フラグは常に (*,*,*,0,0,0,0,0)(0,0) です。(* は任意の値)
(4) 干渉を検知したときに値を算出し、次に干渉を検知するまでその値が保持されます。
(5) ロボット静止時に、他ロボットの移動により干渉を検知した場合は、全軸 0.0 です。
(6) 本変数は動作命令の目的位置を元に動作方向を計算していますので、目的位置に近い位置で衝突した場合は全要素 0.0 になる場合があります。
(7) 読み込み専用です。
(8) 干渉チェックの使用を禁止しているロボットは、常に全軸 0.0 が返されます。
(9) 読み出せる値の単位は P_ColDir と共通です。

P_ColDir

【機能】
衝突検知したときのロボットの動作方向を返します。

【書式】

例) < 位置変数 >=P_ColDir[(< メカ番号 >)]

【用語】
< 位置変数 > 代入する位置変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】
ColChk の文例 2 を参照

【解説】
(1) 衝突検知後の自動復旧動作でロボットの動作方向を確認する際に使用します。
(2) 衝突を検知した瞬間のロボットの動作方向を最大移動軸の値を ±1.0 とした比率で表します。例 : (X 軸方向 :Y 軸方向)=(2:-1) の比率で動作していた場合 … P_ColDir=(1,-0.5,0,0,0,0)(0,0)
(3) 姿勢軸と構造フラグは常に (*,*,*,0,0,0,0,0)(0,0) です。("*" は任意の値)
(4) 衝突を検知したときに値を算出し、次に衝突を検知するまでその値が保持されます。
(5) ロボット静止時に外部から物がぶつかって衝突を検知した場合は、全軸 0.0 となります。
(6) 本変数は動作命令の目的位置を元に動作方向を計算していますので、目的位置に近い位置で衝突した場合は全要素 0.0 になる場合があります。
(7) 読み込み専用です。
(8) 衝突検知の使用を禁止しているロボットは、常に全軸 0.0 が返されます。

【関連命令】
ColChk (コルチェック)、ColLvl (コルレベル)

【関連システム状態変数】
M_ColSts、J_ColMxl

P_CordR

【機能】
共通座標から自ロボットベース座標までの位置を返します。

本機能は対応機種を限定しています。詳細については「5.24 干渉回避機能について」を参照してください。

【書式】

例) < 位置変数 >=P_CordR[(< メカ番号 >)]

【用語】
< 位置変数 > 代入する位置変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】
500 ページの「5.24.10 サンプルプログラム」を参照。

【解説】
(1) 共通座標から自ロボットのベース座標までの位置を参照します。(パラメータ RBCORD の設定値)
(2) パラメータ RBCORD が初期値 (0,0,0,0,0,0) の場合は、すべての座標が 0 で読み出されます。
(3) 読み込み専用です。
(4) ユーザメカでは常に "0" を返します。

P_Curr

【機能】
現在位置 (X,Y,Z,A,B,C,L1,L2)(FL1,FL2) を返します。

【書式】

例) < 位置変数 >=P_Curr[(< メカ番号 >)]

【用語】
< 位置変数 > 代入する位置変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】

```
1 Def Act 1,M_In(10)=1 GoTo *LACT       ' 割り込みを定義
2 Act 1=1                               ' 割り込み有効
3 Mov P1
4 Mov P2
5 Act 1=0                               ' 割り込み無効

100 *LACT
101 P100=P_Curr                         ' 割り込み信号が入った時点での現在位置を取り込みます。
102 Mov P100,-100                       ' P100 の 100mm 上空 ( ツールの Z 方向に -100mm) に移動
103 End                                 ' プログラム終了
```

【解説】
(1) 現在位置を知りたいときに使用します。
(2) 読み込み専用です。

【関連システム状態変数】
J_Curr、P_Fbc

P_CurrR

【機能】
共通座標から見た自ロボットの現在位置を返します。

本機能は対応機種を限定しています。詳細については「5.24 干渉回避機能について」を参照してください。

【書式】

例) < 位置変数 >=P_CurrR[(< メカ番号 >)]

【用語】
< 位置変数 > 代入する位置変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】
500 ページの「5.24.10 サンプルプログラム」を参照。

【解説】
(1) 共通座標から見た自ロボットの現在位置を参照します。
P_Curr をパラメータ RBCORD で変換した値を返します。
(2) 読み込み専用です。
(3) ユーザメカでは常に "0" を返します。

P_ECord

【機能】
Ex-T 制御 /Ex-T スプライン補間命令実行において使用している Ex-T 座標系原点データを返します。

【書式】

例)＜位置変数＞= P_ECord [(＜メカ番号＞)]

【用語】
＜位置変数＞　代入する位置変数を指定します。
＜メカ番号＞　Ex-T 制御 /Ex-T スプライン補間を実行しているメカ番号を設定します。
設定範囲：1 ～ 3
省略時：1
存在しないメカの番号を指定した場合は、実行時にエラー L3870（指定したメカ番号が無効な値です）が発生します。

【文例】
```
1 P1 = P_ECord          ' 現在補間を行っている Ex-T 座標系原点データを位置変数 P1 に代入
```

【解説】
(1) Ex-T 制御 /Ex-T スプライン補間の実行において、最後に使用した Ex-T 座標系原点データを返します。
(2) P_ECord の値をマルチタスクで参照することにより、補間の進行に応じて演算処理や信号出力等の処理を実行することができるようになります。
(3) P_ECord は状況に応じて下表に示す値を返します。
表 4-40：P_ECord が返す値

状況	P_ECord が返す値
電源投入直後	(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)(0,0)
Ex-T 制御 /Ex-T スプライン補間実行中	補間で使用している Ex-T 座標系原点データ
Ex-T 制御 /Ex-T スプライン補間終了後	最後に使用した Ex-T 座標系原点データ
メインプログラムの End 命令実行直後	(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)(0,0)
プログラムリセット操作直後	(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)(0,0)
Ex-T 制御 /Ex-T スプライン補間に対応していないメカ	(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0)(0,0)

(4) 読み込み専用です。

【関連命令】
EMvc（イームーブシー）、EMvr（イームーブアール）、EMvr2（イームーブアール 2）、EMvr3（イームーブアール 3）、EMvs（イームーブエス）、EMvSpl（イームーブスプライン）

P_Fbc

【機能】
サーボからのフィードバック値を元にした現在位置 (X,Y,Z,A,B,C,L1,L2)(FL1,FL2) を返します。

【書式】

例) < 位置変数 >=P_Fbc[(< メカ番号 >)]

【用語】
< 位置変数 > 代入する位置変数を指定します。
< メカ番号 > メカ番号を入れます。1 ～ 3、省略時は 1 です。

【文例】
```
1 P1=P_Fbc                    'P1 にフィードバック現在位置が入ります。
```

【解説】
(1) サーボからのフィードバック値を元にした現在位置を返します。
(2) 読み込み専用です。

【関連命令】
Torq (トルク)

【関連システム状態変数】
P_Curr、J_Fbc/J_AmpFbc、M_Fbd

P_Safe

【機能】
待避点位置（パラメータ JSAFE の直交位置）を返します。

【書式】

例）<位置変数>=P_Safe[(<メカ番号>)]

【用語】
<位置変数>　　代入する位置変数を指定します。
<メカ番号>　　メカ番号を入れます。1～3、省略時は 1 です。

【文例】
1 P1=P_Safe　　'P1 には設定されている待避点位置が入ります。

【解説】
(1) パラメータ JSAFE に登録されている関節位置を直交位置に変換して返します。
(2) 読み込み専用です。

P_Tool/P_NTool

【機能】
ツール変換データを返します。
P_Tool：現在設定されているツール変換データを返します。
P_NTool：初期値 (0,0,0,0,0,0,0,0)(0,0) を返します。

【書式】

例）<位置変数>=P_Tool[(<メカ番号>)]

例）<位置変数>=P_NTool

【用語】
<位置変数>　　代入する位置変数を指定します。
<メカ番号>　　メカ番号を入れます。1～3、省略時は 1 です。

【文例】
1 P1=P_Tool　　'P1 にはツール変換データが入ります。

【解説】
(1) P_Tool は Tool 命令またはパラメータの MEXTL にて設定されたツール変換データを返します。
(2) ツールデータを変更する場合は Tool 命令をご使用ください。
(3) 読み込み専用です。

P_UDev

【機能】

CR7xx-Q シリーズのコントローラでマルチ CPU 間共有メモリへ位置データの書き込み、または参照をおこないます。(CR7xx-Q シリーズコントローラのみ)
位置データ単位（32 ビット x10 分）で書き込み／参照します。

【書式】

参照
<位置変数>=P_UDev(<先頭入出力番号>,<共有メモリアドレス>)

書込み
P_UDev(<先頭入出力番号>,<共有メモリアドレス>)=<位置データ>

【用語】

<位置変数>	代入する位置変数を指定します。
<先頭入出力番号>	CPU ユニットの入出力番号を定数または数値変数で指定します。 (指定する値は先頭入出力番号を 16 進表現し、下位 1 桁を省略した値です) 範囲：16 進表現で &H3E0 ～ &H3E3(10 進表現で 992 ～ 995) 1 号機：&H3E0(10 進表現では 992) 2 号機：&H3E1(10 進表現では 993) 3 号機：&H3E2(10 進表現では 994) 4 号機：&H3E3(10 進表現では 995) 注)データの書き込みは自号機のみが有効です。
<共有メモリアドレス>	CPU ユニットの共有メモリアドレスを定数または数値変数で指定します。 有効範囲は 10000 ～ 24312(10 進数)です。
<位置データ>	書込む位置データを指定します。 定数、変数、論理／算術式、および関数での指定が可能です。

【文例】

```
1 P_UDev(&H3E1, 10010)=P_Curr   '2 号機 CPU の共有メモリアドレス 10010 に現在位置を書き込みます。
2 P1=P_UDev(&H3E2, 10001)       '3 号機 CPU の共有メモリアドレス 10001 の位置データを読み出し、
                                 位置変数 P1 に代入します。
```

【解説】

(1) シーケンサのマルチ CPU 間共有メモリへ位置データの書き込み、または参照をします。
書込むデータ、または参照で返すデータはいずれも位置データです。
(2) 対象の共有メモリは先頭入出力番号と共有メモリアドレスで指定します。
(3) 指定した共有メモリアドレスから 20 ワード(32 ビット x 10)分のデータを対象とします。
(4) 先頭入出力番号は 16 進表現で &H3E0 ～ &H3E3(10 進表現で 992 ～ 995)、共有メモリアドレスは書き込みまたは参照するアドレスが 10 進表現で 10000 ～ 24316 になるように指定ください。
(5) 書き込みは自号機 CPU の共有メモリアドレスのみ可能です。他号機 CPU のアドレスを指定してデータを書込んでも更新されません。
(6) 参照時に対応する共有メモリアドレスがマルチ CPU 高速通信エリア設定（パラメータ：QMLTCPUn）で割付けられていない時は値 0 を返します。

【補足】

表 4-41：<シーケンサ入力信号番号><共有メモリアドレス>に記述できる定数・変数

変数の種類	定数の型			数値変数の型				その他の変数		
	数値注 1) 例)12	2 進数 例)&B1100	16 進数 例)&HC	整数 例)M1%	長精度整数 例)M1&	単精度実数注 1) 例)M1!	倍精度実数注 1) 例)M1#	位置 注 1),注 2) 例)P1.X	関節 注 1),注 2) 例)J1.J1	文字列 例)C1$
使用可否	○	○	○	○	○	○	○	○	×	×

○：使用可能、×：使用不可能

注 1) 実数値は四捨五入されます。

注 2) 変数で扱う数値が角度の場合(位置変数の成分データ A,B,C と関節変数の全成分データ)、設定した値をラジアン単位の数値で格納して処理するため、信号番号の指定が難しくなります。(モニタなどでは度の単位に変換して表示するので設定値と同じ値を表示します)
例)出力信号 8 番を指定するため P1.A に 8 を代入(P1.A=8)しても内部処理は 0.14(rad)を四捨五入して 0 となり、出力信号 0 番を指定することになってしまいます。位置変数の成分データ X,Y,Z は単位が mm のためこのような状態は発生しません。

【参考】号機間データのデータ保証について
自号機の読み出しと、他号機のデータ書き込み / 他号機からの受信タイミングにより、号機ごとのデータに古いデータと新しいデータが混在(データの泣き別れ)することがあります。
CPU 共有メモリを使用したプログラムによる交信において、データの泣き別れを防止する方法を図 4-35 に示します。

1) 32 ビットデータの泣き別れ防止
CPU 共有メモリのユーザー自由エリアへは、偶数アドレス(例:アドレス 10002)を先頭にしてアクセスすることにより 32 ビットデータの泣き別れを防止することができます。

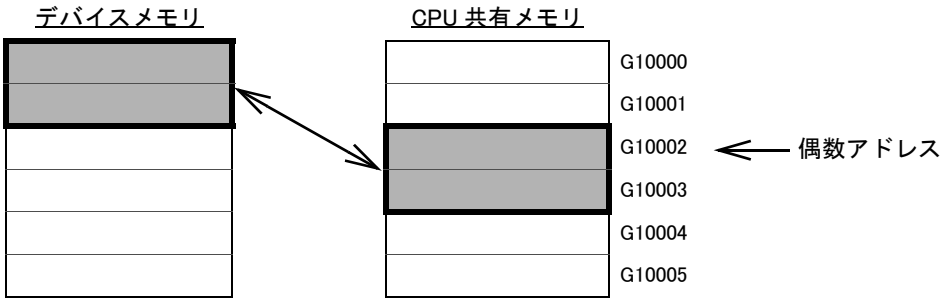


図 4-35:32 ビットデータの泣き別れ防止

2) 32 ビットを超えるデータの泣き別れ防止
プログラムによる読み出しは、ユーザー自由エリアの先頭から順に読み出します。
また書き込み命令ではユーザー自由エリアの最終アドレスから先頭アドレスに向かって、送信データを書き込みます。
そのため交信を行うデータの先頭にインタロック用のデバイスを設けることにより、交信を行うデータの泣き別れを防止することができます。

P_WkCord

【機能】

現在設定されているワーク座標データの参照、および新たなワーク座標の設定をおこないます。
参照、設定の対象はパラメータ：WK1CORD ～ WK8CORD です。
（機能概要につきましては 652 ページの「7.5 Ex-T 制御について」を参照ください。）

【書式】

```
例）<位置変数>=P_WkCord(<ワーク座標番号>)          ’参照
    P_WkCord(<ワーク座標番号>)=<ワーク座標データ>    ’設定
```

【用語】

<位置変数>	代入する位置変数を指定します。
<ワーク座標番号>	1 ～ 8 のワーク座標番号を指定します。 定数、変数、論理 / 算術式、関数が使用可能です。 実数 / 倍精度実数が指定された場合は小数点以下を四捨五入します。
<ワーク座標データ>	位置定数または位置変数でワーク座標データを指定します。 設定する値（座標値）は、ベース座標系から見たワーク座標系原点の位置データです。

【文例】

①既に設定されているワーク座標 1 を元に、新たにワーク座標 2 を設定し、それをワールド座標系とする例

```
1 PW=P_WkCord(1)   ’ワーク座標 1（パラメータ：WK1CORD の設定値）を読み出し PW に代入します。
2 PW.X=PW.X+100    ’ 読み出した X 座標値に 100 を加算します。
3 PW.Y=PW.Y+100    ’ 読み出した Y 座標値に 100 を加算します。
4 P_WkCord(2)=PW   ’ 上記演算結果をワーク座標 2 に設定します。（パラメータ：WK2CORD に設定）
5 Base 2           ’ ワーク座標 2 を新たにワールド座標系とします。
6 Mov P1
```

②PA をワーク座標 4 に設定し、ワーク座標 4 に沿って EX-T 制御直線補間で動作する例

```
1 P_WkCord(4)=PA   ’PA をワーク座標 4 に設定します。（ パラメータ：WK4CORD に設定）
2 EMvs 4,P1        ’ ワーク座標 4 に沿って、Ex-T 制御直線補間で P1 へ移動
```

【解説】

(1) ワーク座標番号を指定して現在設定されているワーク座標値の読み出し、または設定をおこないます。
指定するワーク座標番号 1 ～ 8 は、パラメータ：WK1CORD ～ WK8CORD に対応します。

(2) ワーク座標データの X,Y,Z 成分は、ベース座標系の原点からワーク座標系の原点までの平行移動量を表します。
また、位置データの A,B,C 成分はワーク座標系がロボット座標系に対してどれだけ傾いているかを表します。

X ・・・X 軸方向へ平行移動する距離
Y ・・・Y 軸方向へ平行移動する距離
Z ・・・Z 軸方向へ平行移動する距離
A ・・・X 軸回りに回転させる角度
B ・・・Y 軸回りに回転させる角度
C ・・・Z 軸回りに回転させる角度

A,B,C 成分はワーク座標系原点から正方向にみて、時計回りを正の回転方向にとります。

(3) 構造フラグの内容は意味をもちません。

(4) 本命令でワーク座標の設定をおこなうと、対応するワーク座標番号の WO, WX, WY データ（ワーク座標を教示する 3 点の座標値。パラメータ：WKnWO, WKnWX, WKnWY(n：1 ～ 8)）をクリアします。
例）文例①のステップ 4(P_WkCord(2)=PW) を実行すると、WK2WO, WK2WX, WK2WY には値 "0" が設定されます。

(5) MELFA-BASIC V のみ使用可能です。

(6) Ex-T 制御補間命令では、指定されたワーク座標番号によって、この変数を参照し、Ex-T 制御点とします。よって、P_WkCord への設定は、Ex-T 制御における制御点座標を設定することにもなります。

【関連命令】
Base（ベース）、EMvc（イームーブシー）、EMvr（イームーブアール）、EMvr2（イームーブアール 2）、EMvr3（イームーブアール 3）、EMvs（イームーブエス）

【関連パラメータ】
MEXBSNO、WKnCORD(n:1 ～ 8)、WKnWO, WKnWX, WKnWY(n:1 ～ 8)

P_Zero

【機能】
常に (0,0,0,0,0,0,0,0)(0,0) を返します。

【書式】

例）<位置変数>=P_Zero

【用語】
<位置変数>　　代入する位置変数を指定します。

【文例】
```
1 P1=P_Zero        'P1 には (0,0,0,0,0,0,0,0)(0,0) が入ります。
```

【解説】
(1) P 変数をゼロ初期化したい場合に使用します。
(2) 読み込み専用です。

4.15 関数の詳細説明

4.15.1 記載項目の見方

【機能】　　：関数の機能を示します。
【書式】　　：関数の引数の入力方法を示します。
【文例】　　：関数を使ったプログラム例を示します。
【用語】　　：引数の意味、範囲などを示します。
【解説】　　：詳細な機能や、注意事項を示します。
【参照】　　：関連する関数を示します。
【関連命令】　：関連する命令語を示します。
【関連パラメータ】：関連するパラメータを示します。

4.15.2 各関数の説明

以下に、各関数の説明をアルファベット順に示します。

Abs

【機能】
与えられた値の絶対値を返します。

【書式】

<数値変数>=Abs(<数式>)

【文例】
```
1 P2.C=Abs(P1.C)          'P2.C には P1.C の符号をとった値が入ります。
2 Mov P2
3 M2=-100
4 M1=Abs(M2)              'M1 には 100 が入ります。
```

【解説】
(1) 与えられた値の絶対値(符号を正にした値)を返します。

【参照】
Sgn

ACos

【機能】
指定された余弦値（数値）から逆余弦値（アークコサイン値）を返します。

【書式】

<数値変数>=ACos(<数式>)

【用語】
<数値変数>　指定された数式の値の逆余弦（アークコサイン）の値が計算されて返ります。
出力値の単位はラジアンです。
範囲：0 ～ π
<数式>　余弦値を指定します。
設定範囲：-1.0 ～ +1.0

【文例】
```
1 MRad=ACos(0.6)        '0.6の逆余弦値 (0.927295218001612 rad) をMRadへ代入します。
```

【解説】
(1) 指定された<数式>の逆余弦値を返します。単位はラジアンです。

Align

【機能】

位置の姿勢軸（A，B，C 軸）を最も近い直交姿勢（0，±90，±180）に変換します。
Align は数値の出力のみおこないます、実際の動作は Mov などの動作命令を使用します。

【書式】

< 位置変数 >=Align(< 位置 >)

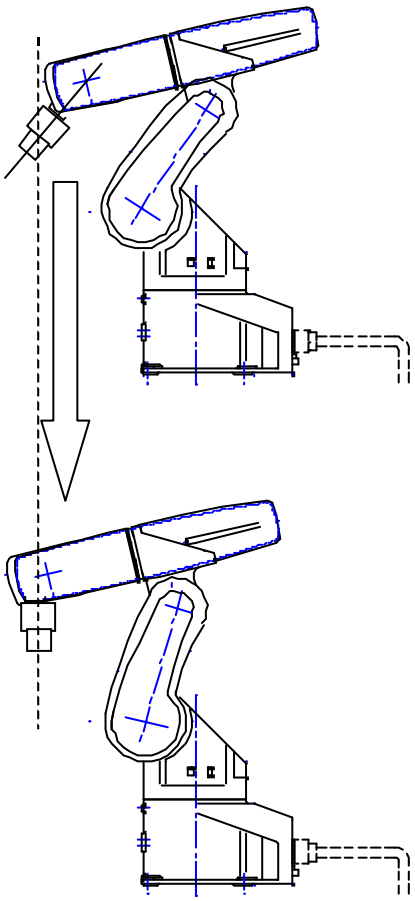
【文例】

```
1 P1=P_Curr
2 P2=Align(P1)
3 Mov P2
```

【解説】

(1) 位置データの A，B，C 成分を最も近い直交姿勢（0，±90，±180）に変換します。
(2) 返る値は位置データのため、左辺が関節変数の場合はエラーになります。
(3) 本機能は垂直多関節の 5 軸ロボットでは使用できません。

下記例は B 軸の例を示しています。



Asc

【機能】
文字列式の最初の文字のキャラクターコードの値を返します。

【書式】

<数値変数>=Asc(<文字列式>)

【文例】
```
1 M1=Asc("A")           'M1 には &H41 が入ります。
```

【解説】
(1) 文字列式の最初の文字のキャラクターコードの値を返します。
(2) 文字列が空文字列の場合はエラーになります。

【参照】
Chr$、Val、Cvi、Cvs、Cvd

ASin

【機能】
指定された正弦値（数値）から逆正弦値（アークサイン値）を返します。

【書式】

```
<数値変数>=ASin(<数式>)
```

【用語】
<数値変数>　指定された数式の値の逆正弦（アークサイン）の値が計算されて返ります。
出力値の単位はラジアンです。
範囲：－π/2 ～ +π/2
<数式>　正弦値を指定します。
設定範囲：-1.0 ～ +1.0

【文例】

```
1 MRad=ASin(-0.4)        '-0.4 の逆正弦値 (-0.411516846067488 rad) を MRad へ代入します。
```

【解説】
(1) 指定された<数式>の逆正弦値を返します。単位はラジアンです。

Atn/Atn2

【機能】
逆正接（アークタンジェント）を計算します。

【書式】

```
<数値変数>=Atn(<数式>)
<数値変数>=Atn2(<数式 1>, <数式 2>)
```

【用語】
<数値変数>　指定された数式による逆正接（アークタンジェント）の値が計算されて返ります。
出力値の単位はラジアンです。
<数式>　ΔY/ΔX の計算値
<数式 1>　ΔY
<数式 2>　ΔX

【文例】

```
1 M1=Atn(100/100)        'M1 には π/4 ラジアンが入ります。
2 M2=Atn2(-100,100)      'M2 には -π/4 ラジアンが入ります。
```

【解説】
(1) 与えられた数式の逆正接を計算します。単位はラジアンです。
(2) 得られる値は、Atn の場合、-π/2<Atn<π/2 です。
(3) 得られる値は、Atn2 の場合、-π<Atn<π です。
(4) Atn2 の場合で<数式 2>が 0 の場合、<数式 1>が正の数の時は π/2 になり、負の数の時は -π/2 になります。
(5) Atn2 の場合、<数式 1>、<数式 2>に引数をもつ関数を記述することできません。実行時にエラーになります。

```
NG 例  M1=Atn2(Max(MA,MB), 100)
       M1=Atn2(CInt(10.2), 100)
```

【参照】
Sin、Cos、Tan

Bin$

【機能】
数値を 2 進数文字列に変換します。

【書式】

< 文字列変数 >=Bin$(< 数式 >)

【文例】
```
1 M1=&B11111111
2 C1$=Bin$(M1)          'C1$ には "11111111" の文字列が入ります。
```

【解説】
(1) 数式の値を 2 進数文字列に変換します。
(2) 数式が、整数でない場合は、小数部を四捨五入した整数値を 2 進数文字列に変換します。
(3) 逆の動作をするコマンドとして Val があります。

【参照】
Hex$、Str$、Val

CalArc

【機能】

指定された 3 点で構成される円弧の情報を取出します。

【書式】

```
<数値変数 4>=CalArc(<位置 1>, <位置 2>, <位置 3>,
                    <数値変数 1>, <数値変数 2>, <数値変数 3>, <位置変数 1>)
```

【用語】

<位置 1>	円弧の始点を指定します。
<位置 2>	円弧の通過点を指定します。MVR 命令の 3 点と同じ意味です。
<位置 3>	円弧の終点を指定します。
<数値変数 1>	指定された円弧の半径が計算されて返ります。(mm)
<数値変数 2>	指定された円弧の中心角が計算されて返ります。(rad)
<数値変数 3>	指定された円弧の円弧長が計算されて返ります。(mm)
<位置変数 1>	指定された円弧の中心座標が計算されて返ります。(位置型、ABC はゼロ)
<数値変数 4>	戻り値。 1：正常に計算ができたとき。 -1：位置 1、位置 2、位置 3 のうち 2 点が同一点、3 点が一直線上の場合。 -2：3 点がほぼ同一点の場合。

【文例】

```
1 M1=CalArc(P1,P2,P3,M10,M20,M30,P10)
2 If M1<>1 Then End                    '異常時は終了
3 MR=M10                               '半径
4 MRD=M20                              '円弧角度
5 MARCLEN=M30                          '円弧長
6 PC=P10                               '中心点座標値
```

【解説】

(1) 指定された位置 1、位置 2、位置 3 の 3 点から生成される円弧の情報を取出します。
(2) 円弧生成が成功して各データが計算された場合の戻り値は 1 になります。
(3) 同一点があったり、3 点が一直線上に並んでいた場合は、戻り値として -1 が返ります。この場合に円弧長は始点から終点への直線の長さ、半径は -1、中心角は 0、中心点は (0,0,0) が返ります。
(4) 円弧生成に失敗した場合は、戻り値として -2 が返ります。円弧が生成できなかった場合は、半径は -1、中心角は 0、円弧長は 0、中心点は (0,0,0) が返ります。
(5) <位置 1>, <位置 2>, <位置 3>, <数値変数 1>, <数値変数 2>, <数値変数 3>, <位置変数 1> に引数をもつ関数を記述することできません。実行時にエラーになります。

Chr$

【機能】
指定した数式の値のキャラクターコードを持つ文字を返します。

【書式】

< 文字列変数 >=Chr$(< 数式 >)

【文例】
```
1 M1=&H40
2 C1$=Chr$(M1+1)                ' C1$ には "A" が入ります。
```

【解説】
(1) 数式の値のキャラクターコードを持つ文字を返します。
(2) 数式の値が、整数でない場合は、小数部を四捨五入した整数値のキャラクターコードを持つ文字を返します。

【参照】
Asc

CInt

【機能】
数式の小数部を四捨五入し、整数に変換します。

【書式】

< 数値変数 >=CInt(< 数式 >)

【文例】
```
1 M1=CInt(1.5)     ' M1 には 2 が入ります。
2 M2=CInt(1.4)     ' M2 には 1 が入ります。
3 M3=CInt(-1.4)    ' M3 には -1 が入ります。
4 M4=CInt(-1.5)    ' M4 には -2 が入ります。
```

【解説】
(1) 数式の値の小数部を四捨五入した値を返します。

【参照】
Int、Fix

CkSum

【機能】
文字列のチェックサムを計算します。

【書式】

<数値変数>=CkSum(<文字列>, <数式 1>, <数式 2>)

【用語】
<文字列>　　チェックサムを計算する文字列を指定します。
<数式 1>　　チェックサムの計算を開始する文字位置を指定します。
<数式 2>　　チェックサムの計算を終了する文字位置を指定します。

【文例】
```
1 M1=CkSum("ABCDEFG",1,3)      'M1 には &H41("A")+&H42("B")+&H43("C")=&HC6 が入ります。
```

【解説】
(1) 文字列中の開始位置から終了位置までの文字列の、各文字の文字コードを加算し、0 ～ 255 の値を返します。
(2) 開始位置が文字列外の範囲を差している場合はエラーになります。
(3) 終了位置が文字列の範囲を超えている場合は、開始位置から文字列の最後の文字までのチェックサムを求めます。
(4) 加算結果が 255 を越えた場合は 255 以下の縮退した値になります。
(5) <文字列>, <数式 1>, <数式 2> に引数をもつ関数を記述することできません。実行時にエラーになります。

Cos

【機能】
余弦（コサイン）を与えます。

【書式】

<数値変数>=Cos(<数式>)

【文例】
```
1 M1=Cos(Rad(60))
```

【解説】
(1) 数式の値の余弦（コサイン）の値を計算します。
(2) 引数の範囲は、数値の値の取りうる範囲全てになります。
(3) 返る値の範囲は -1 ～ 1 になります。
(4) 引数の単位はラジアンです。

【参照】
Sin、Tan、Atn/Atn2

Cvi

【機能】
文字列式の最初の 2 文字のそれぞれの文字コードを数値とみなし整数に変換します。

【書式】

```
< 数値変数 >=Cvi(< 文字列式 >)
```

【文例】
```
1 M1=Cvi("10ABC")              'M1 には &H3031 が入ります。
```

【解説】
(1) 文字列式の最初の 2 文字のそれぞれの文字コードを数値とみなし整数に変換します。
(2) 文字列が 1 文字以下の場合はエラーになります。
(3) 数値を文字列に変換するには MKI$ を使用します。
(4) 外部とのデータ入出力などで数値データを送る時に、データ量を減らす為に用います。

【参照】
Asc、Cvs、Cvd、Mki$、Mks$、Mkd$

Cvs

【機能】
文字列式の最初の 4 文字のそれぞれの文字コードを数値とみなし単精度実数に変換します。

【書式】

```
< 数値変数 >=Cvs(< 文字列式 >)
```

【文例】
```
1 M1=Cvs("FFFF")               'M1 には 12689.6 が入ります。
```

【解説】
(1) 文字列式の最初の 4 文字のそれぞれの文字コードを数値とみなし単精度実数に変換します。
(2) 文字列が 3 文字以下の場合はエラーになります。
(3) 数値を文字列に変換するには Mks$ を使用します。

【参照】
Asc、Cvi、Cvd、Mki$、Mks$、Mkd$

Cvd

【機能】
文字列式の最初の 8 文字のそれぞれの文字コードを数値とみなし倍精度実数に変換します。

【書式】

< 数値変数 >=Cvd(< 文字列式 >)

【文例】
```
1 M1=Cvd("FFFFFFFF")          'M1 には +3.52954E+30 が入ります。
```

【解説】
(1) 文字列式の最初の 8 文字のそれぞれの文字コードを数値とみなし倍精度実数に変換します。
(2) 文字列が 7 文字以下の場合はエラーになります。
(3) 数値を文字列に変換するには MKD$ を使用します。

【参照】
Asc、Cvi、Cvs、Mki$、Mks$、Mkd$

Deg

【機能】
角度単位をラジアン (rad) から度 (deg) に変換します。

【書式】

< 数値変数 >=Deg(< 数式 >)

【文例】
```
1 P1=P_Curr
2 If Deg(P1.C) < 170 Or Deg(P1.C) > -150 Then *NOErr1
3 Error 9100
4 *NOErr1
```

【解説】
(1) 数式の値をラジアン (rad) から度 (deg) に変換します。
(2) 位置データの姿勢角度を位置定数で表す場合 ((500,0,600,180,0,180)(7,0)) の単位は DEG です。P1.C のように位置変数の回転成分を直接参照する場合の単位はラジアン (rad) になります。P1.C の値を DEG で扱うことも可能です。その場合はパラメータ「PRGMDEG」を 1 に設定してください。

【参照】
Rad

Dist

【機能】
2 点間（位置変数）の距離を求めます。

【書式】

<数値変数>=Dist(<位置 1>, <位置 2>)

【文例】
```
1 M1=Dist(P1,P2)                    'M1 には P1 ～ P2 の距離が入ります。
```

【解説】
(1) 位置 1 と位置 2 の距離(mm)を返します。
(2) 位置データの姿勢角度は無視され、X, Y, Z のデータのみを使用して計算されます。
(3) 関節変数は使用できません、実行時にエラーになります。
(4) <位置 1>, <位置 2> に引数をもつ関数を記述することできません。実行時にエラーになります。

Exp

【機能】
指数関数（e を底とする数式のべき乗）を計算します。

【書式】

<数値変数>=Exp(<数式>)

【文例】
```
1 M1=Exp(2)                         'M1 には e^2 が入ります。
```

【解説】
(1) 数式の値の指数関数の値を返します。

【参照】
Ln

Fix

【機能】
数式の整数部を返します。

【書式】

<数値変数>=Fix(<数式>)

【文例】
```
1 M1=Fix(5.5)                    'M1 には 5 が入ります。
```

【解説】
(1) 数式の値の整数部の値を返します。
(2) 数式の値が正の数の場合は、Int と同じ値を返します。
(3) 数式の値が負の数の場合、例えば、Fix(-2.3)=-2.0 になります。

【参照】
CInt、Int

Fram

【機能】
3 つの位置データで指定された座標系（面）を示す位置データを計算します。

【書式】

<位置変数 4>=Fram(<位置 1>, <位置 2>, <位置 3>)

【用語】

<位置 1>	3 つの位置で指定しようとする平面の X，Y，Z の原点となります。変数または定数。
<位置 2>	3 つの位置で指定しようとする平面の X 軸上の点となります。変数または定数。
<位置 3>	3 つの位置で指定しようとする平面の X-Y 平面上の +Y 方向の点となります。変数または定数。
<位置変数 4>	結果を代入する変数。構造フラグは <位置 1> の値を代入します。

【文例】

```
1 Base P_NBase              ' ベース変換データを初期値に戻します。
2 P10=Fram(P1,P2,P3)        'P1,P2,P3 の位置を元に P10 の座標系を作成します。
3 P10=Inv(P10)              'P10 を P10 から見たベース座標系原点位置に変換（逆変換）します。
4 Base P10                  ' ステップ 2 で定義した P10 の位置を新しくワールド座標系に設定します。
      :
```

【解説】
(1) ベース座標系や、基準とする座標系を定義する場合に利用できます。
(2) 3 つの位置変数の X，Y，Z の 3 つの座標値から平面を作成し、原点の位置･平面の傾きを計算して位置変数型として返します。その位置データの X，Y，Z は位置変数 1 の値と同じものに、A，B，C は 3 つの位置で指定される平面の傾きになります。
(3) 引数に関節変数は使用できません。使用した場合はエラーが発生します。
(4) <位置 1>，<位置 2>，<位置 3> に引数をもつ関数を記述することできません。実行時にエラーになります。
NG 例　P10=Fram(FPrm(P01,P02,P03), P04, P05)

【関連パラメータ】
相対変換（演算 *）、434 ページの「5.7 標準ベース座標について」を参照願います。

Hex$

【機能】
数式の値 (-32768 ～ 32767) を 16 進文字列に変換する。

【書式】

```
<文字列変数>=Hex$(<数式>[, <出力文字数>])
```

【文例】

```
10 C1$=Hex$(&H41FF)          'C1$ には"41FF" が入ります。
20 C2$=Hex$(&H41FF,2)        'C2$ には"FF" が入ります。
```

【解説】
(1) 数式の値を 16 進数文字列に変換します。
(2) <出力文字数> を指定すると変換した文字列の右側の文字を指定サイズ分出力します。
(3) 数値が、整数でない場合は、小数部を四捨五入した整数値を 16 進数文字列に変換します。
(4) 逆の動作をするコマンドとして Val があります。
(5) <出力文字数> を指定した場合、<数式> に引数をもつ関数を記述することできません。実行時にエラーになります。
NG 例　C1$=Hex$(Asc("a"),1)

【参照】
Bin$、Str$、Val

Int

【機能】
数式の値を超えない最大の整数を返す。

【書式】

```
<数値変数>=Int(<数式>)
```

【文例】

```
1 M1=Int(3.3)                'M1 には 3 が入ります。
```

【解説】
(1) 数式の値を超えない最大の整数を返します。
(2) 数式の値が正の数の場合は、Fix と同じ値を返します。
(3) 数式の値が負の数の場合、例えば、Fix(-2.3)=-3.0 になります。

【参照】
CInt、Fix

Inv

【機能】
位置変数の逆行列の位置データを求めます。位置の相対演算をする場合に使用します。

【書式】

<位置変数>=Inv(<位置変数>)

【文例】
```
1 P1=Inv(P2)                    'P1 に P2 の逆行列が入ります。
```

【解説】
(1) 位置変数の逆行列の位置データを求めます。
(2) 引数に関節変数は使用できません。使用した場合はエラーが発生します。
(3) 返る値は位置データのため、左辺が関節変数を使用した場合はエラーになります。

JtoP

【機能】
与えられた関節データを位置データに変換します。

【書式】

<位置変数>=JtoP(<関節変数>)

【文例】
```
1 P1=JtoP(J1)                   'P1 には J1(関節型)の位置を直交型で表した場合の位置が入ります。
```

【解説】
(1) 与えられた関節データを、座標変換をおこない、位置データに変換します。
(2) 引数に位置変数は使用できません。使用した場合はエラーが発生します。
(3) 返る値は位置データのため、左辺に関節変数を使用した場合はエラーになります。
(4) 対象とするメカ番号の初期値はメカ 1 となっています。RelM(リリースメカ)命令を実行した後や、マルチタスク制御でスロット 1 以外のプログラムでメカ 1 を対象とする場合、GetM 命令によるメカ番号指定は不要です。他のメカを対象とする場合に GetM 命令を実行してください。

【参照】
PtoJ

Left$

【機能】
文字列の左端から、指定した長さの文字列を取り出します。

【書式】

```
<文字列変数>=Left$(<文字列>, <数式>)
```

【文例】

```
1 C1$=Left$("ABC",2)          'C1$ には "AB" が入ります。
```

【解説】
(1) 文字列の左端から、指定した長さの文字列を取り出します。
(2) 数値が、負の場合や、文字列の長さよりも大きい場合はエラーになります。
(3) <文字列>, <数式>に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Mid$、Right$

Len

【機能】
文字列の長さを返します。

【書式】

```
<数値変数>=Len(<文字列>)
```

【文例】

```
1 M1=Len("ABCDEFG")           'M1 には 7 が入ります。
```

【解説】
(1) 引数の文字列の長さを返します。

【参照】
Left$、Mid$、Right$

Ln

【機能】
自然対数を与えます。(底は e です。)

【書式】

<数値変数>=Ln(<数式>)

【文例】
```
1 M1=Ln(2)                ' M1 には 0.693147 が入ります。
```

【解説】
(1) 数式の値の自然対数を返します。
(2) 数値式がゼロまたは負の場合はエラーになります。

【参照】
Exp、Log

Log

【機能】
常用対数を与えます。(底は 10 です。)

【書式】

<数値変数>=Log(<数式>)

【文例】
```
1 M1=Log(2)               ' M1 には 0.301030 が入ります。
```

【解説】
(1) 数式の値の常用対数を返します。
(2) 数式の値がゼロまたは負の場合はエラーになります。

【参照】
Exp、Ln

Max

【機能】
最大値を求めます。

【書式】

```
<数値変数>=Max(<数式 1>, <数式 2>, …)
```

【文例】

```
1 M1=Max(2,1,3,4,10,100)          'M1 には 100 が入ります。
```

【解説】
(1) 任意の個数の引数の中から最大値を返します。
(2) 一行の登録可能文字数(240 文字)までの範囲内で記述可能です。
(3)<数式 1>, <数式 2>, .. に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Min

Mid$

【機能】
文字列の指定した位置から指定された長さの文字列を返します。

【書式】

```
<文字列変数>=Mid$(<文字列>, <数式 2>, <数式 3>)
```

【文例】

```
1 C1$=Mid$("ABCDEFG",3,2)          'C1$ には "CD" が入ります。
```

【解説】
(1) 第 1 引数の文字列の、第 2 引数で指定した位置から、第 3 引数で指定した長さの文字列を返します。
(2) 数式 2 や数式 3 の数値がゼロまたは負の場合はエラーになります。
(3) 取出す位置の終わりの値が、第一引数の文字列の長さよりも大きい場合はエラーになります。
(4)<文字列>, <数式 2>, <数式 3> に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Left$、Right$、Len

Min

【機能】
最小値を求めます。

【書式】

<数値変数>=Min(<数式 1>, <数式 2>, …)

【文例】
```
1 M1=Min(2,1,3,4,10,100)          'M1 には 1 が入ります。
```

【解説】
(1) 任意の個数の引数の中から最小値を返します。
(2) 一行の登録可能文字数(240 文字)までの範囲内で記述可能です。
(3) <数式 1>, <数式 2>, .. に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Max

Mirror$

【機能】
文字列の各キャラクタコードを 2 進数で表したビット列を反転しキャラクタコード化した文字列を求めます。

【書式】

<文字列変数>=Mirror$(<文字列式>)

【文例】
```
1 C1$=Mirror$("BJ")          ' C1$ には "RB" が入ります。
                             ' "BJ"=&H42,&H4A=&B01000010,&B01001010
                             '  反転 =&H52,&H42=&B01010010,&B01000010
                             '  出力 ="RB"
```

【解説】
(1) 文字列の各キャラクタコードを 2 進数で表したビット列を反転しキャラクタコード化した文字列を求めます。

Mki$

【機能】
数式（整数）の値を 2 バイトの文字列に変換します。

【書式】

```
<文字列変数>=Mki$(<数式>)
```

【文例】

```
1 C1$=Mki$(20299)          'C1$ には"OK"が入ります。
2 M1=Cvi(C1$)              'M1 には 20299 が入ります。
```

【解説】
(1) 数式（整数）の値の下位 2 バイト分を文字列式に変換します。
(2) 文字列を数値に変換するには Cvi を使用します。
(3) 外部との間で数値データを送る時に、データ量を減らす為に用います。

【参照】
Asc、Cvi、Cvs、Cvd、Mks$、Mkd$

Mks$

【機能】
数式（単精度実数）の値を 4 バイトの文字列に変換します。

【書式】

```
<文字列変数>=Mks$(<数式>)
```

【文例】

```
1 C1$=Mks$(100.1)          'C1$ には"100.1"が入ります。
2 M1=Cvs(C1$)              'M1 には 100.1 が入ります。
```

【解説】
(1) 与えられた数式（単精度実数）の値の下位 4 バイト分を文字列式に変換します。
(2) 文字列を数値に変換するには Cvs を使用します。
(3) 外部との間で数値データを送る時に、データ量を減らす為に用います。

【参照】
Asc、Cvi、Cvs、Cvd、Mki$、Mkd$

Mkd$

【機能】
数式（倍精度実数）の値を 8 バイトの文字列に変換します。

【書式】

< 文字列変数 >=Mkd$(< 数式 >)

【文例】
```
1 C1$=Mkd$(10000.1)          'C1$ には"10000.1"が入ります。
2 M1=Cvd(C1$)                'M1 には 10000.1 が入ります。
```

【解説】
(1) 与えられた数式（倍精度実数）の値の下位 8 バイト分を文字列式に変換します。
(2) 文字列を数値に変換するには Cvd を使用します。
(3) 外部との間で数値データを送る時に、データ量を減らす為に用います。

【参照】
Asc、Cvi、Cvs、Cvd、Mki$、Mks$

PosCq

【機能】
与えられた位置が動作範囲内に入っているかどうかを調べます。

【書式】

<数値変数>=PosCq(<位置変数>)

【文例】
```
1 M1=PosCq(P1)                    'P1 の位置が動作範囲内であれば M1 に 1 が入ります。
```

【解説】
(1) 引数で与えられた位置データがロボットの動作範囲に入っているかどうかを調べます。ロボットの動作範囲に入っている場合は 1 が、動作範囲外の場合は 0 が返ります。
(2) 引数は、位置データ型、関節データ型のいずれかを与える必要があります。

PosMid

【機能】
与えられた 2 点間を直線補間した場合の中点位置データを求めます。

【書式】

<位置変数>=PosMid(<位置変数 1>, <位置変数 2>, <数式 1>, <数式 2>)

【文例】
```
1 P1=PosMid(P2,P3,0,0)          'P1 に P2,P3 の中間点の位置データ（姿勢も含め）が入ります。
```

【解説】
(1) 2 つの位置データの間を直線補間した場合の、中点の位置データを求めます。
(2) 第 1 引数は直線補間の始点を与え、第 2 引数は直線補間の終点を指定します。
(3) 第 3 引数、第 4 引数は、Mvs コマンドの Type 引数 2 つに相当します。
(4) 始点と終点の引数は指定した補間タイプで直線補間が可能な位置データである必要があります。例えば、始点と終点の構造フラグが異なる場合はエラーになります。
(5) <位置変数 1>, <位置変数 2>, <数式 1>, <数式 2> に引数をもつ関数を記述することできません。実行時にエラーになります。

PtoJ

【機能】
与えられた位置データを関節データに変換します。

【書式】

<関節変数>=PtoJ(<位置変数>)

【文例】
```
1 J1=PtoJ(P1)              'J1 に P1(直交型位置変数)の値を関節型に変換して入れます。
```

【解説】
(1) 位置データを座標変換し関節データに変換します。
(2) 引数に関節変数(J 変数)は使用できません。使用した場合はエラーが発生します。
(3) 返る値は関節データのため、左辺に位置変数を使用した場合はエラーになります。
(4) 対象とするメカ番号の初期値はメカ 1 となっています。RelM(リリースメカ)命令を実行した後や、マルチタスク制御でスロット 1 以外のプログラムでメカ 1 を対象とする場合、GetM 命令によるメカ番号指定は不要です。他のメカを対象とする場合に GetM 命令を実行してください。

【参照】
JtoP

Rad

【機能】
角度単位 度 (deg) をラジアン (rad) に変換します。

【書式】

<数値変数>=Rad(<数式>)

【文例】
```
1 P1=P_Curr
2 P1.C=Rad(90)
3 Mov P1                   ' 現在位置の C 軸だけを 90 度にした P1 に移動します。
```

【解説】
(1) 数式の値を、角度単位 度(deg)からラジアン(rad)に変換します。
(2) 位置変数の姿勢成分(ABC)へ代入する場合や三角関数を計算する場合によく使用します。

【参照】
Deg

Rdfl1

【機能】
指定した位置の構造フラグを文字データ "R"/"L", "A"/"B", "N"/"F" で返します。

【書式】

< 文字列変数 >=Rdfl1(< 位置変数 >, < 数式 >)

【用語】
< 位置変数 > 構造フラグを取出したい位置変数を指定します。
< 数値 > どの構造フラグを取出したいかを指定します。
0="R"/"L", 1="A"/"B", 2="N"/"F"

【文例】
```
1 P1=(100,0,100,180,0,180)(7,0)     ' 構造ﾌﾗｸﾞ 7(&B111) は RAN なので
2 C1$=Rdfl1(P1,1)                   ' C1$ に "A" が入ります。
```

【解説】
(1) 第 1 引数の位置データの構造フラグのうち、第 2 引数で指定されたものを取出します。
(2) 本関数は、位置データの FL1 の要素から情報を取出します。
(3) < 位置変数 >, < 数式 > に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Rdfl2、Setfl1、Setfl2

Rdfl2

【機能】
指定した関節軸の多回転情報を数値で返します。

【書式】

＜数値変数＞=Rdfl2(＜位置変数＞, ＜数式＞)

【用語】
＜位置変数＞　　多回転情報を取出したい位置変数を指定します。
＜数式＞　　多回転情報を取出したい関節軸の値を指定します。(1 ～ 8)

【文例】
```
1 P1=(100,0,100,180,0,180)(7,&H00100000)                    '
2 M1=Rdfl2(P1,6)                                            'M1 に 1 が入ります。
```

【解説】
(1) 第 1 引数の位置データの多回転情報のうち、第 2 引数で指定された関節軸の値を取出します。
(2) 返される値の範囲は、-8 ～ 7 になります。
(3) 本関数は、位置データの FL2 の要素から情報を取出します。
(4) 構造ﾌﾗｸﾞ 2(多回転情報) は 32 ﾋﾞｯﾄ構造で 1 軸あたり 4 ﾋﾞｯﾄで 8 軸分設定されています
(5) T/B にて表示する場合に多回転がﾏｲﾅｽの場合は -1 ～ -8 を F ～ 8(4 ビット符号付 16 進数表記) にて変換表示しています。
<TB での多回転情報の表示例＞　87654321 軸　　<1 軸あたりの多回転数と表示の関係＞
J6 軸が多回転 +1 の場合 :　FL2=00100000　　・・・・　-2　-1　0　+1　+2　・・・・
J6 軸が多回転 -1 の場合 :　FL2=00F00000　　・・・・　E　F　0　1　2　・・・・
(6)＜位置変数＞, ＜数式＞に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Rdfl1、Setfl1、Setfl2

【関連命令】
JRC（ジョイント・ロール・チェンジ）

Rnd

【機能】
乱数を発生させます。

【書式】

＜数値変数＞=Rnd(＜数式＞)

【用語】
＜数式＞　　乱数の初期値を設定します。0 を設定した場合は乱数の初期値の設定は行わずに、引き続き乱数を取出します。
＜数値変数＞　　0.0 ～ 1.0 の範囲で値を返します。

【文例】

```
1 Dim MRND(10)
2 C1=Right$(C_Time,2)          ' 実行するたびに違う順が得られるように、
3 MRNDBS=Cvi(C1))              ' 時刻値を使って乱数を初期化します。
4 MRND(1)=Rnd(MRNDBS)          ' 乱数の初期値を設定し乱数の一つ目を取出す。
5 For M1=2 To 10               ' 残りの 9 個の乱数を取出す。
6   MRND(M1)=Rnd(0)
7 Next M1
```

【解説】
(1) 引数で与えられた数式の値で乱数を初期化して乱数を取出します。
(2) 引数で与えられた数式の値が 0 の場合は、乱数の初期化は行わずに引き続き乱数を取出します。
(3) 同じ数値で乱数の初期化を行った場合、得られる乱数列は同じ物となります。

Right$

【機能】
文字列の右端から指定された長さの文字列を取り出します。

【書式】

＜文字列変数＞=Right$(＜文字列＞, ＜数式＞)

【文例】

```
1 C1$=Right$("ABCDEFG",3)          'C1$ には "EFG" が入ります。
```

【解説】
(1) 文字列の右端から指定された長さの文字列を取り出します。
(2) 第 2 引数の数値が、負の場合や、第 1 文字列の長さよりも大きい場合はエラーになります。
(3) ＜文字列＞, ＜数式＞に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Left$、Mkd$、Len

Setfl1

【機能】
指定した位置の姿勢フラグを文字（"RAN" など）によって変更します。

【書式】

＜位置変数＞=Setfl1(＜位置変数＞, ＜文字列＞)

【用語】

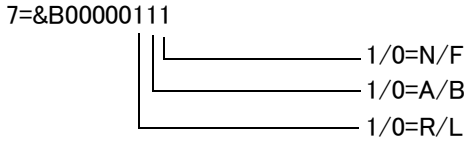
＜位置変数＞	姿勢フラグを変更する位置変数を指定します。
＜文字列＞	変更する姿勢フラグを指定します。複数の指定が可能です。 "R"or"L" : Right/Left の設定 "A"or"B" : Above/Below の設定 "N"or"F" : Nonflip/Flip の設定

【文例】

```
10 Mov P1
20 P2=Setfl1(P1,"LBF")
30 Mov P2
```

【解説】
(1) 第 1 引数の位置データの姿勢フラグを第 2 引数で指定されたものに変更した位置データを返します。
(2) 本関数は位置データの FL1 の要素の情報を変更します。引数で与えられた位置データの内容は変更されません。
(3) 姿勢フラグの指定は、文字列の最後の文字から設定していきます。よって、"LR" などのように指定した場合は結果の姿勢フラグは "L" になります。
(4) 数値で変更する場合は　P1.FL1=7　のように設定してください。
(5) 姿勢フラグの意味はロボットの機種によって異なる場合があります。詳しくは各ロボットの「ロボット本体セットアップから保守まで」編を参照ください。

姿勢フラグは位置定数の (100,0,300,180,0,180)(7,0) の 7 の部分に相当します。実際の位置はビットパターンになっています。



(6) ＜位置変数＞, ＜文字列＞に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Rdfl1、Rdfl2、Setfl2

Setfl2

【機能】
指定した位置の多回転データ（構造フラグ 2）を変更します。

【書式】

<位置変数>=Setfl2(<位置変数>, <数式 1>, <数式 2>)

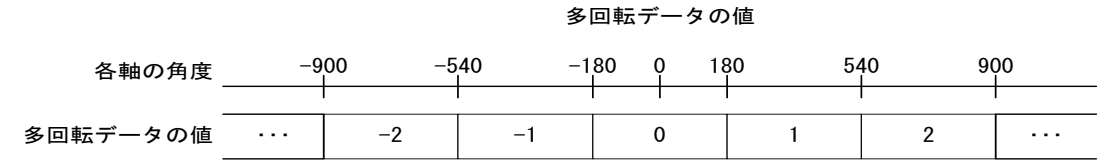
【用語】

<位置変数>	多回転データ（構造フラグ 2）を変更する位置変数を指定します。
<数値 1>	多回転データ（構造フラグ 2）を変更する軸番号を指定します。(1 ～ 8)
<数値 2>	変更する多回転データ値を指定します。(-8 ～ 7)

【文例】

```
10 Mov P1
20 P2=Setfl2(P1,6,1)
30 Mov P2
```

【解説】
(1) 位置データの数式 1 で指定した関節軸の多回転情報を、数式 2 で指定された値に変更した位置データを返します。
(2) 本関数は、位置データの FL2 の要素の情報を変更します。
位置変数の(100, 0, 300, 180, 0, 180)(7, 0) の "0" の箇所に相当します。
(3) 引数で与えられた位置変数の位置の内容(X,Y,Z,A,B,C,FL1)は変更されません。



(4) <位置変数>, <数式 1>, <数式 2> に引数をもつ関数を記述することできません。実行時にエラーになります。

【参照】
Rdfl1、Rdfl2、Setfl1

SetJnt

【機能】
関節変数へ値をセットします。

【書式】

```
<関節変数>=SetJnt(<J1 軸>[, <J2 軸>[, <J3 軸>
                        [, <J4 軸>[, <J5 軸>[, <J6 軸>[, <J7 軸>[, <J8 軸>]]]]]])
```

【用語】
<関節変数>　　変更する関節変数を指定します。
<J1 軸>～<J8 軸>　単位は RAD です(直動軸は mm です)。

【文例】

```
1  J1=J_Curr
2  For M1=0 To 60 Step 10
3   M2=J1.J3+Rad(M1)
4   J2=SetJnt(J1.J1, J1.J2, M2)  'J3 軸の値のみ 10 度ずつ回転させる、J4 軸以降は同じ値
5   Mov J2
6  Next M1
7  M0=Rad(0)
8  M90=Rad(90)
9  J3=SetJnt(M0,M0,M90,M0,M90,M0)
10 Mov J3
```

【解説】
(1) 関節変数の各軸の値を変更できます。
(2) 引数には変数も記述可能です。
(3) J1 軸以外は省略可能です。省略は以降の軸すべて省略となります。(SetJnt(10,10,,,,10) というような記述はできません。)
(4) 引数には、引数を持つ関数を記述することはできません。実行時にエラーになります。

【参照】
SetPos

【関連パラメータ】
AXUNT、PRGMDEG

SetPos

【機能】
位置変数へ値をセットします。

【書式】

```
<位置変数>=SetPos(<X 軸>[, <Y 軸>[, <Z 軸>
                        [, <A 軸>[, <B 軸>[, <C 軸>[, <L1 軸>[, <L2 軸>]]]]]])
```

【用語】
<位置変数>　　変更する位置変数を指定します。
<X 軸>～<Z 軸>　単位は mm です。
<A 軸>～<C 軸>　単位は RAD です。(PRGMDEG パラメータで DEG に切り替え可能です。)
<L1 軸>、<L2 軸>　単位は AXUNT パラメータに依存します。

【文例】

```
1 P1=P_Curr
2 For M1=0 To 100 Step 10
3   M2=P1.Z+M1
4   P2=SetPos(P1.X, P1.Y, M2)       'Z 軸の値のみ 10mm 上昇させる、A 軸以降は同じ値
5   Mov P2
6 Next M1
```

【解説】
(1) 位置変数の各軸の値を変更できます。
(2) 引数には変数も記述可能です。
(3) X 軸以外は省略可能です。省略は以降の軸すべて省略となります。(SetPos(10,10,,,,10) というような記述はできません。)
(4) 引数には、引数を持つ関数を記述することはできません。実行時にエラーになります。

【参照】
SetJnt

【関連パラメータ】
AXUNT、PRGMDEG

Sgn

【機能】
数式の符号を調べます。

【書式】

<数値変数>=Sgn(<数式>)

【文例】
```
1 M1=-12
2 M2=Sgn(M1)        'M2 には -1 が入ります。
```

【解説】
(1) 与えられた数式の値の符号を調べ、以下のような値を返します。

正の数	1
0	0
負の数	-1

Sin

【機能】
正弦（サイン）を計算します。

【書式】

<数値変数>=Sin(<数式>)

【文例】
```
1 M1=Sin(Rad(60))          ' M1 には 0.86603 が入ります。
```

【解説】
(1) 数式の値の正弦（サイン）の値を計算します。
(2) 値の範囲は、数値の値の取りうる範囲全てになります。
(3) 返る値の範囲は -1 ～ 1 になります。
(4) 引数の単位はラジアンです。

【参照】
Cos、Tan、Atn/Atn2

SplECord

【機能】
スプラインファイルに登録されている Ex-T 座標系原点データを位置変数に保存します。

【書式】

<位置変数> = SplECord(<スプライン番号>[, <フレーム変換>])

【用語】

〈位置変数〉	結果を代入する変数です。
〈スプライン番号〉	取得したい経路点の情報を保持しているスプラインファイルの番号を､定数または数値変数で指定します。 設定範囲:1～99
〈フレーム変換〉	フレーム変換の実行内容を､定数または数値変数で指定します。 設定範囲: 0:フレーム変換を実行しません。 1:スプラインファイルに設定された座標系を使ってフレーム変換を実行します。 2:SetCalFrm命令で設定された座標系を使ってフレーム変換を実行します。 省略時:フレーム変換を実行しません。

【文例】
```
1 P_WkCord(1)=SplECord(8)     ' スプラインファイル 1 の経路点 5 番の位置データを位置変数 P1 に代入
2 P1=SplECord(5)              ' スプラインファイル 5 に登録されている Ex-T 座標系原点データを位置変数 P1 に代入
3 P2=SplECord(10,1)           ' スプラインファイル 10 に登録されている Ex-T 座標系原点データに対しフレーム変換したデータを P2 に代入。
```

【解説】
(1) <スプライン番号> で指定されたスプラインファイルに登録されている Ex-T 座標系原点データを返します。
(2) <スプライン番号> に対応したスプラインファイルがコントローラに保存されていない場合は、エラー L2610(スプラインファイルを開くことができません) が発生します。

(3)＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」または「2(SetCalFrm 命令で設定した座標系を使用して実行)」を指定した場合は、指定された方法に基づいて Ex-T 座標系原点データを変換し、補正後のデータを返します。
(4)＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」を指定しているにも関わらず、スプラインファイルに座標系が設定されていない場合は、エラー L2042(フレーム変換座標が設定されていません)が発生します。
(5)＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」を指定している場合に座標系が算出できない場合は、エラー L2041(フレーム変換座標が算出できません)が発生します。
(6)＜フレーム変換＞に「2(SetCalFrm 命令で設定した座標系を使用して実行)」を指定すると、最後に実行した SetCalFrm 命令により設定された座標系を使ってフレーム変換を実行します。一度も SetCalFrm 命令が実行されておらず座標系が設定されていない場合は、エラー L2042(フレーム変換座標が設定されていません)が発生します。
(7) スプラインファイルに Ex-T 座標設定がされていない場合は、エラー L2610(Ex-T 座標設定不正)が発生します。

SplPos

【機能】
スプラインファイルに登録されている任意の経路点データを位置変数に保存します。

【書式】

＜位置変数＞ = SplPos(＜スプライン番号＞,＜経路点番号＞[,＜フレーム変換＞])

【用語】

〈位置変数〉　結果を代入する変数です。構造フラグは指定のスプラインファイルに登録されている経路点データの値を代入します。

〈スプライン番号〉　取得したい経路点の情報を保持しているスプラインファイルの番号を、定数または数値変数で指定します。
設定範囲:1～99

〈経路点番号〉　取得したい経路点の番号を、定数または数値変数で指定します。
設定範囲:1～5000

〈フレーム変換〉　フレーム変換の実行内容を、定数または数値変数で指定します。
設定範囲:
0:フレーム変換を実行しません。
1:スプラインファイルに設定された座標系を使ってフレーム変換を実行します。
2:SetCalFrm命令で設定された座標系を使ってフレーム変換を実行します。
省略時:フレーム変換を実行しません。

【文例】

```
1 P1=SplPos(1,5)        ' スプラインファイル 1 の経路点 5 番の位置データを位置変数 P1 に代入
2 P2=SplPos(1,10,1)     ' スプラインファイル 1 の経路点 10 番の位置データに対しフレーム変換したデータを位置変数 P2 に代入
```

【解説】
(1)＜スプライン番号＞で指定されたスプラインファイルに登録されている＜経路点番号＞に対応した経路点の位置データを返します。
(2) 構造フラグの値はスプラインファイルに登録されている経路点データの値となります。
(3)＜スプライン番号＞に対応したスプラインファイルがコントローラに保存されていない場合は、エラー L2610(スプラインファイルを開くことができません)が発生します。
(4)＜経路点番号＞に対応した経路点情報がスプラインファイルに登録されていない場合は、エラー L2611(経路点データ登録されていません)が発生します。
(5)＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」または「2(SetCalFrm 命令で設定した座標系を使用して実行)」を指定した場合は、指定された方法に基づいて経路点データを変換し、その経路点を返します。

(6)＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」を指定しているにも関わらず、スプラインファイルに座標系が設定されていない場合は、エラー L2042(フレーム変換座標が設定されていません)が発生します。
(7)＜フレーム変換＞に「1(スプラインファイルに設定した座標系を使用して実行)」を指定している場合に座標系が算出できない場合は、エラー L2041(フレーム変換座標が算出できません)が発生します。
(8)＜フレーム変換＞に「2(SetCalFrm 命令で設定した座標系を使用して実行)」を指定すると、最後に実行した SetCalFrm 命令により設定された座標系を使ってフレーム変換を実行します。一度も SetCalFrm 命令が実行されておらず座標系が設定されていない場合は、エラー L2042(フレーム変換座標が設定されていません)が発生します。

SplSpd

【機能】
スプライン補間命令 (MvSpl、EMvSpl 命令) にてエラーとならずに指定可能な最大速度 [mm/s] を算出します。

【書式】

＜数値変数＞= SplSpd(＜スプライン番号＞)

【用語】
〈数値変数〉　結果を代入する変数です。
〈スプライン番号〉　取得したい経路点の情報を保持しているスプラインファイルの番号を､定数または数値変数で指定します。
設定範囲:1～99

【文例】

```
1 MS=SplSpd(8)      ' スプラインファイル 8 の経路点データを用いてスプライン補間を実行した際にエラーとならずに指定が可能な最大速度を MS に代入。
2 MvSpl 8,MS,50     ' スプラインファイル 8 の経路点を MS で設定された速度で通過するスプライン補間を実行
```

【解説】
(1)＜スプライン番号＞で指定されたスプラインファイルに登録されている経路点データからスプライン命令 (MvSpl、EM v Spl 命令) 実行時にエラーとならずに指定可能な最大速度を算出します。
(2) 算出される最大速度はプログラムオーバーライドを 100% とした際の速度となります。
(3) 算出した速度を使用してスプライン補間を実行した場合は、ロボットの姿勢が大きく変化すると速度オーバーが発生する可能性があります。
(4)＜スプライン番号＞に対応したスプラインファイルがコントローラに保存されていない場合は、エラー L2610(スプラインファイルを開くことができません)が発生します。

Sqr

【機能】
数式の平方根を計算します。

【書式】

<数値変数>=Sqr(<数式>)

【文例】
```
1 M1=Sqr(2)                ' M1 には 1.41421 が入ります。
```

【解説】
(1) 数式の値の平方根を計算します。
(2) 引数で与えられる数式の値が負の数の場合はエラーが発生します。

Strpos

【機能】
文字列内から指定した文字列を検索します。

【書式】

<数値変数>=Strpos(<文字列 1>, <文字列 2>)

【文例】
```
1 M1=Strpos("ABCDEFG","DEF")        'M1 には 4 が入ります。
```

【解説】
(1) 第 1 引数の文字列の中から、第 2 引数で指定された文字列が最初に現れる位置を返します。
(2) 文字列 2 の長さが 0 の場合はエラーになります。
(3) 例えば、文字列 1 が "ABCDEFG" で、文字列 2 が "DEF" の場合は、4 が返ります。
(4) 文字列が見つからなかった場合は、0 が返ります。
(5) <文字列 1>, <文字列 2> に引数をもつ関数を記述することできません。実行時にエラーになります。

Str$

【機能】
数式の値を 10 進文字列に変換します。

【書式】

<文字列変数>=Str$(<数式>)

【文例】
```
1 C1$=Str$(123)                'C1$ には "123" が入ります。
```

【解説】
(1) 数式の値を 10 進数文字列に変換します。
(2) 逆の動作をするコマンドとして Val があります。

【参照】
Bin$、Hex$、Val

Tan

【機能】
正接（タンジェント）を計算します。

【書式】

<数値変数>=Tan(<数式>)

【文例】
```
1 M1=Tan(Rad(60))              ' M1 には 1.73205 が入ります。
```

【解説】
(1) 数式の値の正接（タンジェント）の値を返します。
(2) 引数で与える値の範囲は、数値の値の取りうる範囲全てになります。
(3) 返る値の範囲は、数値の値の取りうる範囲全てになります。
(4) 引数の単位はラジアンです。

【参照】
Sin、Cos、Atn/Atn2

Val

【機能】
文字列の値を数値に変換します。

【書式】

<数値変数>=Val(<文字列式>)

【文例】
```
1 M1=Val("15")
2 M2=Val("&B1111")
3 M3=Val("&HF")
```

【解説】
(1) 与えられた文字列式を数値に変換します。
(2) 文字列には、2 進数(&B)、10 進数、16 進数(&H)が使用できます。
(3) 上記の文例では、M1,M2,M3 は同じ値(15)になります。

【参照】
Bin$、Hex$、Str$

Zone

【機能】

指定した位置が指定領域内 (2 点で指定される直方体) に入っているかをチェックします。

【書式】

< 数値変数 >=Zone(< 位置 1>, < 位置 2>, < 位置 3>)

【用語】

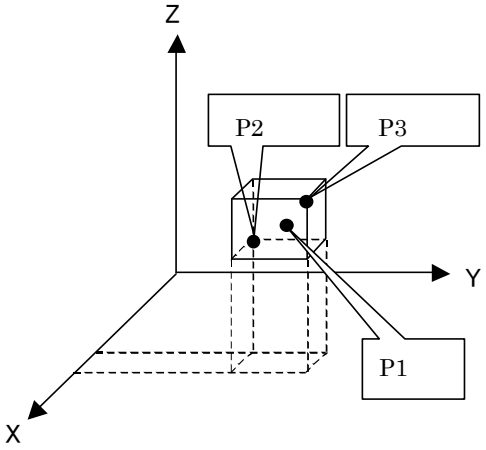
< 位置 1>　　チェックしたい位置
< 位置 2>　　領域を指定する 1 点目の位置
< 位置 3>　　領域を指定する 2 点目の位置 (対角点)
位置 1 ～ 3 は直交座標系の変数 (P 変数　X,Y,Z,A,B,C,L1,L2) を設定します。

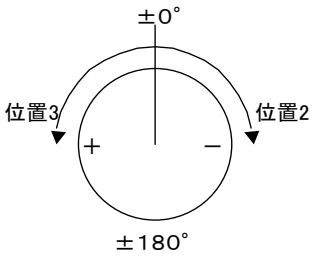
【文例】

```
1 M1=Zone(P1,P2,P3)
2 If M1=1 Then Mov P_Safe Else End
```

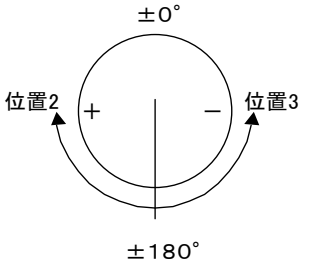
【解説】

(1) 位置 1 が、位置 2 と位置 3 の 2 点で指定される直方体 (2 点が立方体の対角点になります) の空間に入っているかどうかのチェックをし、入っている場合は 1 を、入っていない場合は 0 を返します。
(2) 空間内に入っているかどうかは位置 1 の全ての要素 (X,Y,Z,A,B,C,L1,L2) の値が、位置 2 と位置 3 の値の間に存在するかどうかをチェックします。
(3) 姿勢角 (A,B,C) については、位置 2 の角度から、正の方向に回転して、位置 3 の角度に到達するまでの範囲に入っているかどうかをチェックします。例)P2.A が -100 P3.A が +100 の場合 P1.A が 50 なら範囲内となります。B,C 軸も同様のチェックをします (下図参照)。
(4) チェックしない、または、存在しない位置成分につきましては、単位が deg の場合、位置 2 に -360、位置 3 に 360、単位が mm の場合、位置 2 に -10000、位置 3 に 10000 を設定してください。
(5) < 位置 1>, < 位置 2>, < 位置 3> に引数をもつ関数を記述することできません。実行時にエラーになります。





例)-90～0～+90の場合は
<位置2>のABCに－側の値を入れる。
<位置3>のABCに＋側の値を入れる。



例)-90～180～+90の場合は
<位置2>のABCに＋側の値を入れる。
<位置3>のABCに－側の値を入れる。

Zone2

【機能】
指定した位置が指定領域内（2 点で構成される円筒体）に入っているかどうかをチェックします。

【書式】

< 数値変数 >=Zone2(< 位置 1>, < 位置 2>, < 位置 3>, < 数式 >)

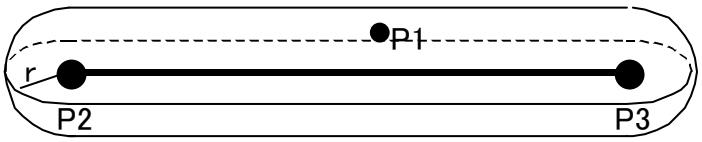
【用語】

< 位置 1>	チェックしたい位置
< 位置 2>	領域を指定する 1 点目の位置
< 位置 3>	領域を指定する 2 点目の位置
< 数式 >	両端の半球の半径

【文例】

```
1 M1=Zone2(P1,P2,P3,50)
2 If M1=1 Then Mov P_Safe Else End
```

【解説】
(1) 位置 1 が、位置 2 と位置 3 の 2 点および数式で指定される半径で構成される空間（下図参照）に入っているかのチェックをし、入っている場合は 1 を、入っていない場合は 0 を返します。
(2) この関数はチェック位置（X, Y, Z の各座標）が指定空間内には入っているかをチェックするもので姿勢成分については考慮しません。



(3) < 位置 1>, < 位置 2>, < 位置 3>, < 数式 > に引数をもつ関数を記述することできません。実行時にエラーになります。

Zone3

【機能】
指定した位置が指定領域内（3 点で構成される平面を基にした直方体）に入っているかどうかをチェックします。

【書式】

< 数値変数 >=Zone3(< 位置 1>,< 位置 2>,< 位置 3>,< 位置 4>,< 数式 W>, < 数式 H>,< 数式 L>)

【用語】

< 位置 1>	チェックしたい位置
< 位置 2>	領域を指定する 1 点目の位置
< 位置 3>	領域を指定する 2 点目の位置
< 位置 4>	< 位置 2>、< 位置 3> とともに、領域を構成する平面を指定する点の位置
< 数式 W>	領域を構成する直方体の幅 [mm]
< 数式 H>	領域を構成する直方体の高さ [mm]
< 数式 L>	領域を構成する直方体の < 位置 2>、< 位置 3> からのそれぞれの奥行き [mm]

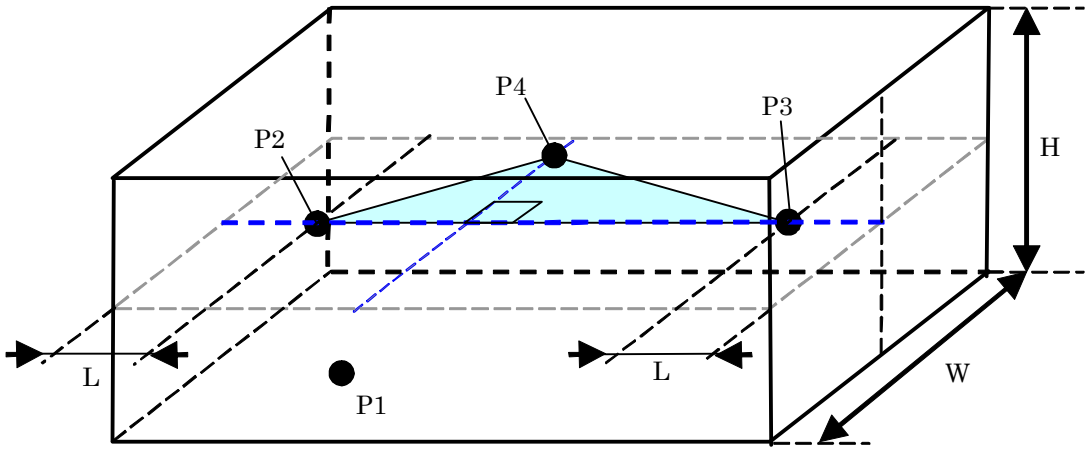
【文例】

```
1 M1=Zone3(P1,P2,P3,P4,100,100,50)
2 If M1=1 Then Mov P_Safe Else End
```

【解説】

1) 位置 1 が、位置 2、位置 3、位置 4 の 3 点および数式 W、数式 H、数式 L で構成される空間（下図参照）に入っているかのチェックを行い、入っている場合には 1 を、入っていない場合には 0 を返します。

2) この関数は、チェックする位置である＜位置 1>(X、Y、Z の各座標）が指定空間内に入っているかをチェックするもので、姿勢成分については考慮しません。



3) ＜位置 1> ～＜位置 4>、＜数式 W> ～＜数式 L> に引数をもつ関数を記述することはできません。実行時にエラーになります。

4) ＜数式 W>、＜数式 H> に負の値が入力された場合はエラーが発生します。

5) ＜位置 2> ～＜位置 4> に同一の位置、または同一直線上の位置が入力された場合、指定領域を作成できないため、チェックを行わずに -1 を返します。

＜数式 L> が負の数で、絶対値が＜位置 2> と＜位置 3> の間の距離の半分を下回る場合は、チェックを行わずに -1 を返します .

5 パラメータでの設定機能

本コントローラには、表 5-1 などに示す各種パラメータがあります。パラメータの設定を変更することによりいろいろな機能や初期設定を変更することが可能です。

No.	分類	内容	参照ページ
1	動作パラメータ	ロボットの動作範囲、座標系、ハンド関連の設定用のパラメータです。	408 ページ
2	信号パラメータ	信号関連用のパラメータです。専用信号に関しては、509 ページの「6.2 シーケンサリンク入出力機能」を参照願います。	419 ページ
3	操作パラメータ	コントローラ、T/B 等の操作に関するパラメータです。	425 ページ
4	言語パラメータ	ロボット言語に関するパラメータです。	427 ページ
5	通信パラメータ	通信に関するパラメータです。	430 ページ

専用入出力信号のパラメータに関しては 509 ページの「6.2 シーケンサリンク入出力機能」を参照してください。
パラメータを変更した場合は必ずコントローラの電源を再投入 (OFF → ON) してください。電源を入れ直さないとパラメータの設定内容が有効になりません。パラメータの詳細な操作方法は、本書 /85 ページの「3.14 パラメータ画面の操作」を参照願います。

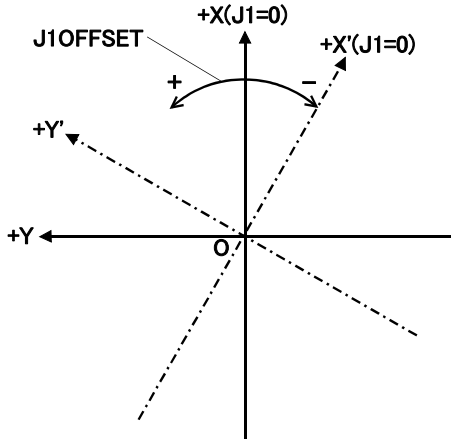
⚠注意　パラメータの変更は、その機能と設定値を十分に確認した上でおこなってください。そうでない場合は、ロボットが予期せぬ動きをする場合があり人身事故や物損の原因になります。

5.1 動作パラメータ

ロボットの動作範囲、座標系、ハンド関連の設定用のパラメータです。

表 5-1：動作関連パラメーター覧表

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
関節動作範囲	MEJAR	実数 16	関節座標系のオーバーランリミット値を指定します。 各軸の動作範囲を設定します。 動作範囲を広げることは推奨しません。メカストッパにぶつかる可能性があります。 注)垂直5軸タイプのロボットで、J1オフセット角度(J1OFFSET)を設定した場合は、J1軸の関節動作範囲を変更することはできませんのでご注意ください。 マイナス、プラスの2方向(-J1,+J1,-J2,+J2,.....-J8,+J8) 単位：deg	メカ毎の設定値
直交動作範囲	MEPAR	実数 6	直交座標系のオーバーランリミット値を指定します。 ロボットの動作範囲を直交座標系にて制限をかけます。 ロボットを装置内に設置し、ﾏﾆｭｱﾙ操作時に周辺装置にぶつけないようにしたい場合に設定します。 マイナス、プラスの2方向(-X,+X,-Y,+Y,-Z,+Z) 単位：mm	(-X,+X,-Y,+Y,-Z,+Z)= -10000,10000, -10000,10000, -10000,10000
狭角広角ﾘﾐｯﾄ機能 ※RV-4F/7F/13F/20F シリーズのみ	MELTEXS	整数 1	ロボットアーム先端がベースに干渉しないよう保護するための狭角広角リミット機能の有効/無効を切替えます。 ベースに近い箇所にアーム先端を動作させる必要がある場合は、本パラメータに"0"を設定してください。この場合、アーム先端とベースが干渉しないように十分ご注意願います。 (狭角広角リミット領域については「別冊/標準仕様書」に記載の動作範囲図を参照) 0：無効 1：有効	1(有効) RV-4F/7F/13F/20Fシリーズのみ。 他の機種は0(無効)

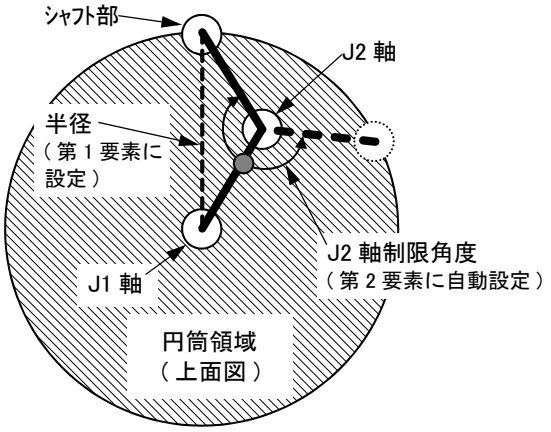
パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
J1 軸ｵﾌｾｯﾄ角度	J1OFFSET	実数 2	垂直5軸タイプのロボットの、J1軸のオフセット角度を設定します。 他のタイプのロボットでは機能しません。 注)J1OFFSETを設定するとJ1=0度の向きを変更するため、自動的に関節動作範囲(MEJAR)を補正します。また、J1OFFSETを設定した場合、J1軸の関節動作範囲を変更することはできませんのでご注意ください。 第1要素:本機能の有効/無効を指定します。機能を有効とする場合は1.0を設定し、無効とする場合は0.0を設定してください。 第2要素:J1=0度(X軸)の向きを、このパラメータに設定した角度分だけオフセットさせることができます。(単位:度) 設定範囲:-360≦第2要素≦+360  	0.0, 0.0
標準ﾂｰﾙ座標 「4.5 ロボットの座標系解説」、「5.6 標準ツール座標について」を参照願います。	MEXTL	実数 6	ハンド先端(制御点)とメカニカルインタフェース(ハンド取り付け面)との関係について初期値を設定します。 出荷時設定はメカニカルインタフェースが制御点になっています。 ハンドを取り付け制御点をハンドの先端にしたい場合は変更してください。(直交またはツールジョグ時にハンド先端で姿勢制御できます) (X,Y,Z,A,B,C) 単位:XYZはmm、ABCはdeg	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
ﾂｰﾙ座標 1 ～ 16 「M_Tool」を参照願います。	MEXTL1 : MEXTL16	実数 6	M_TOOL変数に1～16を代入するとそれぞれのパラメータ値でツールデータを切り替えできます。	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
標準ﾍﾞｰｽ座標 「4.5 ロボットの座標系解説」、「5.7 標準ベース座標について」を参照願います。	MEXBS	実数 6	ワールド座標系とベース座標系の位置関係を設定します。 出荷時設定はワールド座標系とベース座標系は一致しています。 装置全体の座標系として変更する場合に設定します。このパラメータはあまり変更する必要はありません。装置全体の座標系を統一したい場合などに設定します。 (X,Y,Z,A,B,C) 単位:XYZはmm、ABCはdeg 注)プログラムが運転中もしくは中断中には値を変更できません	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
ﾍﾞｰｽ座標番号 「4.5 ロボットの座標系解説」を参照願います。	MEXBSNO	整数 1	ベース座標番号を指定してワールド座標系を設定します。(ベース変換) また、現在の設定状態を表示します。 設定値の内容: 0: P_NBase(システム初期値)を指定(P_NBase=0, 0, 0, 0, 0, 0のためベース変換をクリアします) 1～8:ワーク座標系1～8(パラメータ:WK1CORD～WK8CORDの値)の設定値を指定 -1: Base命令でベース変換データを直接指定した、または標準ベース座標パラメータ:MEXBSで直接設定した。 (注:設定値-1は読み出しのみ有効です) 注)プログラムが運転中もしくは中断中には値を変更できません	-1

<table>
<tr><th>パラメータ</th><th>パラメータ名</th><th>配列数
文字数</th><th>内容説明</th><th>出荷時設定値</th></tr>
<tr><td rowspan="3">ユーザ定義領域

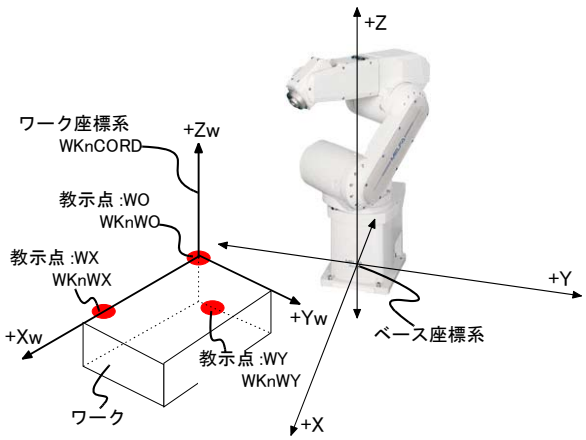
「5.8 ユーザ定義領域について」を参照願います。</td><td colspan="3">ユーザ定義領域の設定(最大32領域)とその領域内でのロボットの挙動を指定します。</td><td></td></tr>
<tr><td>AREA*CS
＊は 1 ～ 32</td><td>整数 1</td><td>ユーザ定義領域*の座標系を設定します。(*=1～32)

0:ワールド座標系(従来互換)
1:ベース座標系</td><td>0</td></tr>
<tr><td>AREA*P1
＊は 1 ～ 32</td><td>実数 8</td><td>ユーザ定義領域*の対角点1の位置座標、及び姿勢データ・付加軸座標を設定します。第1要素からX、Y、Z、A、B、C、L1、L2の順番で定義されます。
<table><tr><th></th><th>内容</th><th>単位</th></tr><tr><td>X,Y,Z 成分</td><td>対角点 1 の位置座標を設定</td><td>mm</td></tr><tr><td>A,B,C 成分</td><td>姿勢領域を設定</td><td>deg</td></tr><tr><td>L1,L2 成分</td><td>付加軸領域を設定</td><td>mm、deg</td></tr></table>
〈注意〉
・AREA*CSで指定した座標系での値を設定してください。
・姿勢チェックを行わない場合は、A,B,C座標を-360に設定してください。
・付加軸を用いる場合は、L1.L2成分を設定してください。
・X,Y,Z,L1,L2成分については、AREA*P2と入替えて設定しても、定義される領域は変わりません。</td><td>(X, Y, Z, A, B, C, L1, L2) =
0.0, 0.0, 0.0, −360.0, −360.0, −360.0 ,0 ,0</td></tr>
<tr><td>AREA*P2
＊は 1 ～ 32</td><td>実数 8</td><td>ユーザ定義領域*の対角点2の位置座標、及び姿勢データ・付加軸座標を設定します。第1要素からX、Y、Z、A、B、C、L1、L2の順番で定義されます。
<table><tr><th></th><th>内容</th><th>単位</th></tr><tr><td>X,Y,Z 成分</td><td>対角点 2 の位置座標を設定</td><td>mm</td></tr><tr><td>A,B,C 成分</td><td>姿勢領域を設定</td><td>deg</td></tr><tr><td>L1,L2 成分</td><td>付加軸領域を設定</td><td>mm、deg</td></tr></table>
〈注意〉
・AREA*CSで設定した座標系での値を設定してください。
・姿勢チェックを行わない場合は、A,B,C座標を+360に設定してください。
・付加軸を用いる場合は、L1.L2成分を設定してください。
・X,Y,Z,L1,L2成分については、AREA*P1と入替えて設定しても、定義される領域は変わりません。</td><td>(X, Y, Z, A, B, C, L1, L2) =
0.0, 0.0, 0.0, +360.0, +360.0, +360.0 ,0 ,0</td></tr>
<tr><td rowspan="2">ユーザ定義領域

「5.8 ユーザ定義領域について」を参照願います。</td><td>AREA*ME
＊は 1 ～ 32</td><td>整数 1</td><td>ユーザ定義領域*を有効にするメカ番号を指定します。
メカ番号の設定は、1～3で通常は1を設定します。
0:無効
1:メカ1(通常設定)
2:メカ2
3:メカ3
(0設定の場合は、領域チェックを行ないません)</td><td>0</td></tr>
<tr><td>AREA*AT
＊は 1 ～ 32</td><td>整数 1</td><td>ユーザ定義領域内にロボットが入った場合の挙動を指定します
0:無効
1:信号出力(専用出力・状態変数出力)
2:エラー出力
〈設定内容〉
<table><tr><th>設定</th><th>領域内</th><th>領域外</th></tr><tr><td rowspan="2">信号出力</td><td>専用出力信号 ON
(※1)</td><td>専用出力信号
OFF</td></tr><tr><td>状態変数の対応 Bit ON
(※2)</td><td>状態変数の対応 Bit OFF</td></tr><tr><td>エラー出力</td><td>エラー出力による停止
(H2090 エラー発生)</td><td>−</td></tr></table>※1:システム状態変数(M_Uar32、M_Uar)
※2:専用入出力の信号番号はUSRAREAで設定します
〈注意〉
エラー出力を選択した場合、姿勢領域チェックと付加軸チェック領域は無視され、位置領域のみでチェックします。</td><td>0(無効)</td></tr>
</table>

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
自由平面ﾘﾐｯﾄ 「5.9 自由平面リミット」を参照願います。	自由平面で設定するオーバーランリミットです。 3点の座標から平面を作成し、原点を含まない側の領域を動作範囲外とします。以下の3種類のパラメータを使用し、最大8つのリミットの設定が出来ます。			
	SFC*P ＊は 1 ～ 8	実数 9	平面を作成する為の3点を指定します。 X1,Y1,Z1：平面の原点位置 X2,Y2,Z2：平面のX軸上の位置 X3,Y3,Z3：平面のX-Y平面上の+Y方向の位置	(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3)=0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
	SFC*ME ＊は 1 ～ 8	整数 1	自由平面リミットを有効にするメカ番号を指定します。 メカ番号の設定は1～3で通常は1を設定します。	0
	SFC*AT ＊は 1 ～ 8	整数 1	設定した自由平面リミットの有効/無効を指定します。 0：無効(初期値) 1：有効(動作可能領域はロボット座標原点側) -1：有効(動作可能領域はロボット座標原点が無い側)	0(無効)
円筒ﾘﾐｯﾄ ※ 以下の S/W Ver と機種でのみ有効。 S/W Ver. F-Q ｼﾘｰｽﾞ：R4c 以降 F-D ｼﾘｰｽﾞ：S4c 以降 機種： RH-3FHR ｼﾘｰｽﾞ	J1軸を中心とした円筒領域で動作制限をかけることができます。 J1軸を中心とした半径をパラメータに設定します。			
	MECAR	実数 2	パラメータの構成：(半径, J2軸制限角度) 第1要素に半径を設定します。単位：mm(小数点以下2桁) 半径が0の場合，本機能は働きません。 半径がマイナス値，又はアームを伸ばしても届かない距離の場合、入力時にエラーとなります。 半径で定義される円筒ﾘﾐｯﾄはJ2軸の動作を制限させます。J2軸の制限角度は自動計算され，絶対値で第2要素に入ります。(読み出しのみで変更はできません) ※関節動作範囲ﾊﾟﾗﾒｰﾀ(MEJAR)と比較してより小さいほうで動作制限がかかります。  注) 本機能はRH-3FHRｼﾘｰｽﾞに機種を限定しています。	0, 0
待避点位置	JSAFE	実数 8	待避点位置を指定します。 ロボットプログラムより Mov P_SAFE と実行するか、外部信号のSAFEPOS信号を入力することによって待避点位置へ動作します。 (J1,J2,J3,J4,J5,J6,J7,J8) 単位：deg	機種によって異なります。
ﾒｶｽﾄｯﾊﾟ原点	MORG	実数 8	メカストッパ原点位置を指定します。 (J1,J2,J3,J4,J5,J6,J7,J8) 単位：deg	機種によって異なります。
ﾕｰｻﾞ指定原点	USERORG	実数 8	ユーザ指定原点位置を指定します。通常は設定する必要はありません。 (J1,J2,J3,J4,J5,J6,J7,J8) 単位：deg	機種によって異なります。
特異点近傍警告機能選択 「5.17 特異点近傍警告について」を参照願います。	MESNGLSW	整数 1	特異点近傍警告の有効/無効を指定します。 (無効/有効=0/1) 本パラメータが"有効"に設定されている場合は、パラメータ：BZR(ブザーのON/OFF)が"OFF"に設定されていても本警告音は鳴ります。	1(有効)

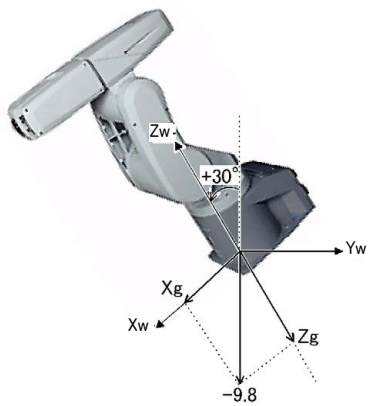
パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ｼﾞｮｸﾞ設定	JOGJSP	実数 3	関節ジョグの送り量およびステップ運転の速度を指定します。 (定寸H, 定寸L, 最大オーバーライド) 定寸H:ジョグ速度High表示の時の送り量 単位:deg 定寸L:ジョグ速度Low表示の時の送り量 単位:deg 最大ｵｰﾊﾞｰﾗｲﾄﾞ:OPｵｰﾊﾞｰﾗｲﾄﾞ×最大ｵｰﾊﾞｰﾗｲﾄﾞで動作します。	メカ毎の設定値
	JOGPSP	実数 3	直交ジョグの送り量およびステップ運転の速度を指定します。 (定寸H, 定寸L, 最大オーバーライド) 定寸H:ジョグ速度High表示の時の送り量 単位:mm 定寸L:ジョグ速度Low表示の時の送り量 単位:mm 最大ｵｰﾊﾞｰﾗｲﾄﾞ:OPｵｰﾊﾞｰﾗｲﾄﾞ×最大ｵｰﾊﾞｰﾗｲﾄﾞで動作します。 最大速度は250mm/sを超えての動作はできません。	メカ毎の設定値
ｼﾞｮｸﾞ速度規制値	JOGSPMX	実数 1	"MANUAL"モード時のロボット動作速度を制限します。単位:mm/s 250より大きな値を設定しても、最大値は250に制限されます。	250.0

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ﾜｰｸ座標系	WKnCORD n は 1 ～ 8	実数 6	ワーク座標の座標値です。 (X, Y, Z, A, B, C) 単位: mm, または度 ワークジョグ操作での基準座標や、ワーク座標データとして使用します。 ソフトウェアバージョンR5(F-Qシリーズ)/S5(F-Dシリーズ)以降では、Ex-T制御における制御点(Ex-T座標)としても使用します。 Ex-T制御については、652ページの「7.5 Ex-T制御について」を参照してください。 ワーク座標データとして使用する場合、ロボット機種によって有効な軸成分が異なります。434ページの「5.7 標準ベース座標について」を参照してください。 T/Bでのワーク座標設定(定義)操作で定義された座標値が設定されますが、本パラメータに座標値を入力することでもワーク座標を定義することができます。この場合，下記の教示によってワーク座標を定義するための3点(パラメータ:WKnWO，WKnWX，WKnWY(nは1～8))の各座標値は0にクリアします。  注)ワーク座標の教示は、ベース変換が初期値の状態(ワールド座標系とベース座標系が一致している状態)でおこなったほうが管理がしやすくなります。特に、ワーク座標を複数定義される場合は混乱防止のために推奨します	(0.00, 0.00, 0.00, 0.00, 0.00, 0.00)
	WKnWO n は 1 ～ 8	実数 3	ワーク座標を定義するための教示位置で，ワーク座標の原点の位置です。(T/Bでの教示操作の"WO"に対応します。上図参照) (X, Y, Z) 単位: mm 注)この座標値を入力しただけではワーク座標は定義されません。T/Bワーク座標設定画面の[定義]操作またはRT ToolBox2のワーク座標パラメータ画面の[書き込み]操作でワーク座標の計算処理をおこなってください。	(0.00, 0.00, 0.00)
	WKnWX n は 1 ～ 8	実数 3	ワーク座標を定義するための教示位置で，ワーク座標の+X軸上の位置です。(T/Bでの教示操作の"WX"に対応します。上図参照) (X, Y, Z) 単位: mm 注)この座標値を入力しただけではワーク座標は定義されません。T/Bワーク座標設定画面の[定義]操作またはRT ToolBox2のワーク座標パラメータ画面の[書き込み]操作でワーク座標の計算処理をおこなってください。	(0.00, 0.00, 0.00)
	WKnWY n は 1 ～ 8	実数 3	ワーク座標を定義するための教示位置で，ワーク座標のX-Y平面上での+Y軸側の位置です。(T/Bでの教示操作の"WY"に対応します。上図参照) (X, Y, Z) 単位: mm 注)この座標値を入力しただけではワーク座標は定義されません。T/Bワーク座標設定画面の[定義]操作またはRT ToolBox2のワーク座標パラメータ画面の[書き込み]操作でワーク座標の計算処理をおこなってください。	(0.00, 0.00, 0.00)
ワークジョグ動作モード ※ 以下の S/W Ver でのみ有効。 S/W Ver. F-Q ｼﾘｰｽﾞ : R5 以降 F-D ｼﾘｰｽﾞ : S5 以降	WKnJOGMD n は 1 ～ 8	整数 1	ワークジョグ操作における動作モードをワーク座標ごとに設定します。 0:ワークジョグ(A,B,Cの動作は、ワーク座標のX軸、Y軸、Z軸と平行な軸回りに回転します。制御点の位置は変わりません。) 1:Ex-Tジョグ(A,B,Cの動作は、ワーク座標のX軸、Y軸、Z軸を中心として制御点を移動しながら回転します。)	0

<table>
<tr><th>パラメータ</th><th>パラメータ名</th><th>配列数
文字数</th><th>内容説明</th><th>出荷時設定値</th></tr>
<tr><td>中断時ｼﾞｮｸﾞ送り後の自動復帰設定

「5.10 中断時のジョグ送り後の自動復帰設定」を参照願います。</td><td>RETPATH</td><td>整数 1</td><td>プログラム運転中、停止によってプログラムを中断したあとにジョグ動作等で移動させた場合、再始動時に中断位置に戻ってからプログラムの再始動をする設定にします。
機能を無効にした場合は現在位置から次のポイントまで設定されている動作命令で移動します。中断点にはもどりません。
0:無効
1:関節補間で戻る
2:直線補間で戻る
注) 直線補間で戻る場合は、3軸直交補間で近回り動作をします。
注) 円弧補間命令とMva命令では、0と設定されていても1と同じ動作となります。</td><td>1</td></tr>
<tr><td>重力方向

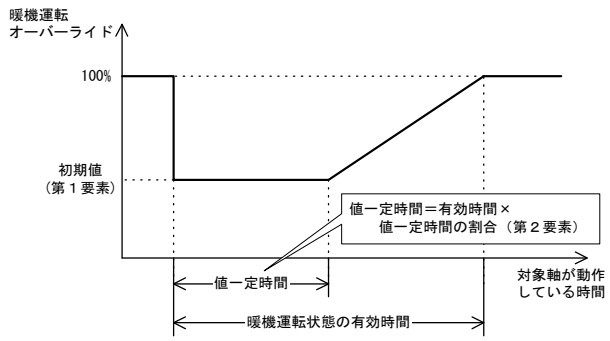
※RV-F シリーズのみ</td><td>MEGDIR</td><td>実数 4</td><td>据付姿勢に応じて、ロボットに作用する重力加速度の方向と、大きさをワールド座標系の X 軸、Y 軸、Z 軸ごとに設定します。(単位 : mm/ 秒 2)
要素数は 4 個で、左から順に据付姿勢、X 軸の重力加速度、Y 軸の重力加速度、Z 軸の重力加速度です。
<table><tr><th>据付姿勢</th><th>設定値
(据付姿勢 , X 軸の重力加速度 ,
Y 軸の重力加速度 , Z 軸の重力加速度)</th></tr><tr><td>床置き (標準)</td><td>(0.0, 0.0, 0.0, 0.0)</td></tr><tr><td>壁掛け</td><td>(1.0, 0.0, 0.0, 0.0)</td></tr><tr><td>天吊り注 1)</td><td>(2.0, 0.0, 0.0, 0.0)</td></tr><tr><td>任意の姿勢注 2)</td><td>(3.0, ***, ***, ***)</td></tr></table>注 1) RH-3FHR シリーズは第 1 要素は "0.0" のままでご使用願います。
注 2) *** には、数値が入ります。
任意の姿勢の設定値については、下例を参照ください。

重力加速度の設定例を以下に示します。
例) ロボットが前方に 30 度傾いた姿勢の場合 (下図参照)
X 軸方向重力加速度 (Xg) = 9.8 x sin(30 度) = 4.9
Z 軸方向重力加速度 (Zg) = 9.8 x cos(30 度) = 8.5
ただし、ワールド座標系の Z 軸とは方向が逆なので、-8.5 とする。
Y 軸方向重力加速度 (Yg) = 0.0
よって、設定値は (3.0, 4.9, 0.0, -8.5) となります。
  </td><td>0.0, 0.0, 0.0, 0.0</td></tr>
<tr><td>ﾊﾝﾄﾞ初期状態

「5.13 ハンド初期状態について」を参照願います。</td><td>HANDINIT</td><td>整数 8</td><td>電源投入時のエアハンドI/Fの出力を設定します。
本パラメータはロボットの先端にあるハンド専用信号900番台)の電源ON時の初期値を指定するものです。
ハンド制御を汎用IO(10000～18191番台以外)で制御する場合(HANDTYPEパラメータにて900番以外の信号を指定)の電源ON時の初期状態を設定するには、このHANDINITパラメータではなく、ORS*パラメータを使用してください。
このORS*パラメータで設定された値が電源ON時の信号の初期値になります。</td><td>1,0,1,0,1,0,1,0</td></tr>
</table>

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ﾊﾝﾄﾞﾀｲﾌﾟ 「5.12 ハンドタイプについて」を参照願います。	HANDTYPE	文字列 8	シングル/ダブルソレノイドのハンドタイプと、出力信号番号を設定します(D:ダブルソレノイド、S:シングルソレノイド)。 ハンドタイプに続けて信号番号を設定します。 D900の場合、信号番号900と901に信号出力されます。 D(ダブルソレノイド)の場合は信号がかさならないように設定してください。	CR750/CR751: D900,D902,D904,D906 ,,,, CR760: D764,D766,D768,D770 ,,,,
ﾊﾝﾄﾞ入出力ﾀｲﾌﾟ	HIOTYPE	整数 1	内蔵エアハンドインタフェースの電気的仕様(ソース/シンク)を設定します。ご使用になるソレノイドバルブなどの仕様に合わせて設定してください。 0:ソース/1:シンク	1
ﾊﾝﾄﾞ、ﾜｰｸ条件 (最適加減速 , 衝突検知で使用) 「5.16 ハンド、 ワーク条件設定 (最適加減速設定) について」を参照願います。	プログラムでOadl On時のハンド条件・ワーク条件を設定します。 最大8まで設定でき、どの条件の組み合わせを使用するかは、LoadSet命令によって選択できます。 注) ハンド、ワーク条件は正しく設定してください。実際の搭載負荷を下回る設定をおこなった場合、ロボットに使用している機構部品の寿命が短くなる可能性があります。			
	HNDDAT0	実数 7	ハンドの初期条件を設定します。(ツール座標系で指定します。) 電源ON直後はこの設定値が採用されます。 ジョグ操作時に衝突検知を使用される場合は実際のハンド条件を設定してお使いください。設定されない場合誤検知する可能性があります。 (重量, 大きさX, 大きさY, 大きさZ, 重心X, 重心Y, 重心Z) 単位:Kg, mm	機種によって異なります。
	HNDDAT* ＊は 1 ～ 8	実数 7	ハンドの条件を設定します。(ツール座標系で指定します。) (重量, 大きさX, 大きさY, 大きさZ, 重心X, 重心Y, 重心Z) 単位:Kg, mm	標準負荷 ,0.0,0.0,0.0,0.0,0.0,0.0,0.0
	WRKDAT0	実数 7	ワークの初期条件を設定します。(ツール座標系で指定します。) 電源ON直後はこの設定値が採用されます。 (重量, 大きさX, 大きさY, 大きさZ, 重心X, 重心Y, 重心Z) 単位:Kg, mm	0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
	WRKDAT* ＊は 1 ～ 8	実数 7	ワークの条件を設定します。(ツール座標系で指定します。) (重量, 大きさX, 大きさY, 大きさZ, 重心X, 重心Y, 重心Z) 単位:Kg, mm	0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
	HNDHOLD* ＊は 1 ～ 8	整数 2	HOPEN*やHCLOSE*を実行した時に、ワークの把持/未把持を設定します。 (OPEN時の設定, CLOSE時の設定) (未把持/把持=0/1)	0,1
最適加減速設定 「5.16 ハンド、 ワーク条件設定 (最適加減速設定) について」を参照願います。	ACCMODE	整数 1	最適加減速モードの電源ON直後の有効/無効を設定します。 (無効/有効=0/1)	1
最適加減速補正率	JADL	実数 8	最適加減速時の加減速補正率の初期値(電源投入時の値)を設定します(%)。最適加減速制御にて計算される加減速度に対してかけられる割合です。本設定値を大きくすることにより高速動作が可能ですが、そのまま一定時間動作させると過負荷エラーやオーバーヒートエラーが発生する可能性があります。エラーがでる場合は設定値を小さくしてください。 初期状態として過負荷エラーやオーバーヒートエラーが発生しにくい状態に初期値を設定してあります。 加速度、減速度両方にかかります。 ※過負荷エラーとは 高速回転での熱によるモータ破損を防ぐため、負荷率が一定以上になった場合にでるエラーです。 ※オーバーヒートエラーとは 高速回転での熱による位置検出器の損傷を防ぐため、温度が一定以上に上昇した場合にでるエラーです。	機種によって異なります。

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
加減速度最適化ﾊﾟﾀｰﾝ選択 ※RH-F シリーズのみ（RH-3FHR シリーズは除く）	MAPMODE	整数 1	RH-Fｼﾘｰｽﾞ(RH-3FHRｼﾘｰｽﾞは除く)において、ｼｬﾌﾄ(J3軸)の高さに応じた加減速度最適化ﾊﾟﾀｰﾝを、標準加減速度ﾊﾟﾀｰﾝとするか、又は高速加減速度ﾊﾟﾀｰﾝとするかを選択します。 0:標準加減速度ﾊﾟﾀｰﾝ/1:高速加減速度ﾊﾟﾀｰﾝ 初期設定は標準加減速度ﾊﾟﾀｰﾝで、ｼｬﾌﾄ(J3軸)先端の振動(含む残留振動)を最小限に抑えるよう働いています。お客様でのﾛﾎﾞｯﾄによる作業内容において、この振動が悪影響を与えない範囲で高加減速度ﾊﾟﾀｰﾝを選択してﾛﾎﾞｯﾄを高速に動作させることができます。ﾊﾟﾀｰﾝの詳細は、別冊の「標準仕様書」を参照願います。	0
最適速度制御 調整係数 ※ 以下の S/W Ver でのみ有効。 S/W Ver. F-Q ｼﾘｰｽﾞ：R5 以降 F-D ｼﾘｰｽﾞ：S5 以降	OPTOVC	実数 1	最適速度制御の調整係数を設定します。 設定範囲:0.30～1.00 最適速度制御とは、Spd命令による速度指定をおこなわない直線補間動作/円弧補間動作にて、最適な速度で動作する機能です。設定値を大きくすることによりロボットを高速に動作させることができますが、特異点近傍など、ロボットの動作姿勢の変化が大きい場合に速度リミットオーバーエラーが発生しやすくなります。 また、速度リミットオーバーエラーが発生する場合に、本設定値を下げることによってエラーの発生を抑制することができますが、ロボットの動作速度は遅くなります。 ※速度リミットオーバーエラーとは エラー番号H213nのエラーです。(nは軸番号1～8を示します) モータへの速度指令が許容値を超える場合にこのエラーが発生します。	機種によって異なります。
速度調整補間機能 ｽｲｯﾁ ※ 以下の機種でのみ有効。 機種：RH-3FHR ｼﾘｰｽﾞ	SPDOPT	整数 1	速度調整補間機能の有効/無効を設定します。 1：有効(電源投入時、速度調整補間機能有効となります) 0：無効(電源投入時、速度調整補間機能無効となります) 1 または0 の時、プログラム中でSpdOpt 命令を使って速度調整補間機能の有効/無効を切り替えることができます。 注) 本機能はRH-3FHRｼﾘｰｽﾞに機種を限定しています。	RH-3FHR シリーズ 1 その他の機種 0
衝突検知	COL	整数 3	衝突検知の使用可 / 不可と電源 ON 直後の有効 / 無効を定義します。 要素 1：衝突検知機能の使用可 (1)/ 不可 (0) 要素 2：自動運転時初期状態として有効 (1)/ 無効 (0) 要素 3：ジョグ操作中有効にする (1)/ しない (0)/NOERR モード (2) NOERR モードは衝突を検知してもエラーを表示しません。サーボオフするのみとなります。衝突を検知した時にエラーが頻繁にでて操作しづらい場合にご利用ください。 自動運転 (ポジションジャンプ、ステップ送り含む) 以外の時は、ジョグ操作用 (要素 3) の設定に依存します。	RH-3FH/6FH/12FH/20FHシリーズ: 1,0,1 RV-Fシリーズ: 0,0,1 RH-3FHRシリーズ: 1,1,1
検知ﾚﾍﾞﾙ	COLLVL	整数 8	自動運転時における各軸の検知レベル (感度) の初期値を関節軸毎に設定します。 設定範囲：1 ～ 500　単位：% ※ 設定範囲外のときは、最も近い設定範囲内の値として処理します。	機種により異なります。
ｼﾞｮｸﾞ時の検知ﾚﾍﾞﾙ	COLLVLJG	実数 8	ジョグ操作時 (中断中を含む) の検知レベル (感度) を関節軸毎に設定します。単位(%) 検知レベル (感度) を上げる場合は数値を小さくします。 ジョグ操作中衝突していないのに衝突検知エラーがでる場合は数値を大きくするよう調整してください。 設定範囲：1 ～ 500　単位：% ※ 設定範囲外のときは、最も近い設定範囲内の値として処理します。	機種により異なります。

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
手首回転角(A 軸)座標系選択	RCD	整数 1	垂直 5 軸タイプのロボットの手首回転角(直交座標 A 軸)の制御と表示方式を切り替えます。他のタイプのロボットでは機能しません。 2: 一般角方式 動作の前後で A 軸の値が同じであればハンドの姿勢を保持するように A 軸を制御します。ただし、手首の傾き(直交座標 B 軸)によってはハンドの姿勢を保持できない場合があります。通常は出荷時の設定を変更せず、この方式をご使用ください。	2(一般角方式)
暖機運転ﾓｰﾄﾞ設定	WUPENA	整数 1	暖機運転モードの有効 / 無効を指定します。 0 : 無効 1 : 有効 注)上記以外の値を設定した場合はすべて無効となります。 注)マルチメカの場合、メカ毎の設定となります。	0(無効)
暖機運転ﾓｰﾄﾞ対象軸	WUPAXIS	整数 1	暖機運転モードで制御の対象となる関節軸を、ビットの ON/OFF による 16 進数(下位ビットより J1,J2,…)で指定します。 ビット ON : 対象軸 ビット OFF : 非対象軸 低温下での稼働時に誤差過大エラーとなる関節軸を対象軸とします。 注)存在しない軸のビットを ON しても対象軸とはなりません。 注)対象軸がない場合は暖機運転モードは無効となります。 注)マルチメカの場合メカ毎の設定となります。	0
暖機運転ﾓｰﾄﾞ制御時間	WUPTIME	整数 2	暖機運転モードの処理で使用する時間を指定します。 (有効時間,再開時間)単位 : 分 有効時間 : 暖機運転状態となり速度を下げて動作する時間を指定します。(設定範囲 : 0 ～ 60) 再開時間 : 暖機運転状態が解除された後、対象軸が停止し続けた場合に、再び暖機運転状態となるまでの時間を指定します。(設定範囲 : 1 ～ 1440) 注)設定範囲外の値を指定した場合は最も近い設定範囲の値として処理します。 注)有効時間が 0 分の場合は暖機運転モードは無効となります。 注)マルチメカの場合メカ毎の設定となります。	1,60
暖機運転ｵｰﾊﾞﾗｲﾄﾞ	WUPOVRD	整数 2	暖機運転状態における速度に関する設定をおこないます。(初期値,値一定時間の割合)単位 : % 初期値 : 暖機運転状態のときに動作速度に掛かるオーバーライド(暖機運転オーバーライド)の初期値を指定します。(設定範囲 : 50 ～ 100) 値一定時間の割合 : 暖機運転状態のときに動作速度に掛かるオーバーライドが初期値から変化しない時間を、有効時間に対する割合で指定します。(設定範囲 : 0 ～ 50) 暖機運転オーバーライドの値と各要素の設定値の対応は、下図のとおりです。  注)設定範囲外の値を指定した場合は、最も近い設定範囲の値として処理します。 注)オーバーライドの初期値が 100% の場合は、暖機運転モードは無効となります。 注)マルチメカの場合、メカ毎の設定となります。	70, 50

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ｺﾝﾌﾟﾗｲｱﾝｽｴﾗｰの機能設定	CMPERR	整数 1	本パラメータの設定により、エラー 2710 ～ 2740（コンプライアンス制御で生成される位置指令が異常の場合に発生するエラー）を発生させなくすることができます。 1：エラー発生有効 0：エラー発生無効 対象となるエラーの内容は次のとおりです。 2710：本来の位置指令からの変位が大きすぎる。 2720：コンプライアンス指令の関節リミットオーバー 2730：コンプライアンス指令の速度オーバー 2740：コンプライアンス指令の座標変換エラー これらのエラーが発生する場合は、コンプライアンス制御が正常に機能していない状態ですので、本来は教示位置やプログラムの内容を見直してエラーが発生しないようにする必要があります。本パラメータの 0（エラー発生無効）への変更は、エラーによって運転が中断しなくても運用上問題がないと判断できる場合のみとしてください。	1 （エラー発生有効）
Cmp Jnt 用電流制限レベル	CMPJCLL	整数 1	関節座標系のコンプライアンスモード（Cmp Jnt 命令）における、各軸モータの電流制限レベルを変更します。 設定範囲：1（制限高）～ 10（制限低） 関節座標系のコンプライアンスモードで動作しているときに誤差過大 1 エラー（H096n）が発生する場合、本パラメータの設定値を大きくすると、誤差過大 1 エラーの発生が抑えられます。 （変更後の設定値は、次回の Cmp Jnt 命令実行から反映されます。）	1
過負荷検知ﾚﾍﾞﾙの最適化	OLTMX	整数 1	ロボット使用環境での周囲温度の上限値を設定します。 この設定値に基づきロボット動作時の過負荷検知レベルを最適化します。（単位：℃） 設定範囲：0 ～ 40 本パラメータに関する機能の詳細を 479 ページの「5.22 過負荷レベルの最適化について」に記載します。	RH-3FH/6FH/12FH/20FH：40 RV-2Fｼﾘｰｽﾞ：40 RV-4F/7F/13F/20F/50Fｼﾘｰｽﾞ、RH-3FHR：30
ﾊﾞｯﾃﾘ消耗時間ｴﾗｰ発生時間間隔	ITBATERR	整数 1	バッテリ消耗時間エラーの発生時間間隔を指定します。 （単位：時間） 設定範囲：1 ～ 336。 設定値が 1 より小さい時は 1 に、336 より大きいときは 336 と見なします。	24
ｼｰｹﾝｻとの時刻同期（CR7xx-Q のみ）	TIMESYNC	整数 1	ﾛﾎﾞｯﾄｺﾝﾄﾛｰﾗの設定時刻とシーケンサの設定時刻を同期させるか否かを選択します。 （同期する / しない =1/0）	0
WTHIF 命令仕様	WTHFUNC	整数 1	WthIf 命令の停止タイプを指定します。 0：停止タイプ 1 1：停止タイプ 2 停止タイプについては、Def Act 命令の説明を参照ください。	0
Cnt 無効時の動作命令終了条件指定	MVTERM	整数 1	Cnt 無効時の動作命令終了条件を指定します。 F シリーズ（RV-2F/2FL、RH-3FHR は除く）では、Cnt0 時の目的位置到達性を向上するため動作命令の完了条件を変更しています。F シリーズ以前の機種（SD/SQ シリーズ）で作成したプログラムをそのまま F シリーズ（RV-2F/2FL、RH-3FHR は除く）で使用すると、動作時間が長くなる場合があります。動作速度を優先する場合は本パラメータを変更し、位置決めが必要な位置では Fine P や Dly 命令などを使用してください。 0：速度指令出力完了 1：Cnt 1 相当 2：速度指令生成完了	0：Fシリーズ （RV-2F/2FL、RH-3FHRは除く） 2：RV-2F/2FL、RH-3FHR、SD/SQシリーズ
退避点復帰モード	ESCMODE	整数 1	退避点復帰時、現在位置の設定をツール定義位置にするか、メカニカルインタフェース原点にするかを選択します。 0：TCP 位置（ツール定義位置）に設定 1：メカニカルインタフェース原点に設定	0
コントローラ CPU 運転モード	DRVMODE	整数 1	ソフトウェアを使用したシミュレーション時に、ロボットとの接続が必要か否かを指定します。 F-Q シリーズの場合は、ドライブユニットとの接続が必要か否かを指定します。 0：通常モード （ロボットまたはドライブユニットとの接続が必要） 1：ロボットサーボ切り離しモード （ロボットまたはドライブユニットとの接続が不要）	0

5.2 信号パラメータ

信号関連用のパラメータです。専用信号に関しては 516 ページの「6.3 専用入出力」を参照願います。

表 5-2：信号関連パラメーター覧表

<table>
<tr><th>パラメータ</th><th>パラメータ名</th><th>配列数
文字数</th><th>内容説明</th><th>出荷時設定値</th></tr>
<tr><td>専用入出力信号</td><td></td><td></td><td>専用入出力信号のパラメータに関しては、516ページの「6.3専用入出力」を参照してください。</td><td></td></tr>
<tr><td>始動信号の入力時に数値入力からプログラム番号を読み込む</td><td>PST</td><td>整数 1</td><td>通常外部入力信号からプログラムを選択するには数値入力信号(IODATA)にプログラム番号を設定し、プログラム選択信号(PRGSEL)で番号を確定し、始動信号で動作させます。本機能を有効にすると、プログラム選択信号が不要になり、始動信号(START)がONしたタイミングで数値入力信号(IODATA)らプログラム番号を読み込みます。
(機能無効/有効=0/1)</td><td>0(無効)</td></tr>
<tr><td>停止入力 B 接点指定</td><td>INB</td><td>整数 1</td><td>停止入力信号のA接点/B接点を切り替えます。
(A接点/B接点=0/1)
切替の対象は，専用入力信号:STOP，STOP2とコントローラ専用停止入力:SKIPです。
※SKIPについての詳細は別冊の「標準仕様書」、または「特殊仕様書」を参照願います。</td><td>0(A接点)</td></tr>
<tr><td>ロボットエラー出力</td><td>ROBOTERR</td><td>整数 1</td><td>EMGOUTコネクタのロボットエラー出力端子を開放するエラーレベルを設定します。
工場出荷時設定では、どのエラーレベルが発生しても開放するようになっています。
例えば、警告レベルを無視して、LowレベルまたはHighレベルのどちらかまたは両方が発生したとき、同出力端子を開放するようにするためには、3 を設定します。

<table>
<tr><th rowspan="2">設定値</th><th colspan="3">発生エラ－レベル</th></tr>
<tr><th>警告</th><th>Low</th><th>High</th></tr>
<tr><td>0</td><td>－</td><td>－</td><td>－</td></tr>
<tr><td>1</td><td>－</td><td>－</td><td>開放</td></tr>
<tr><td>2</td><td>－</td><td>開放</td><td>－</td></tr>
<tr><td>3</td><td>－</td><td>開放</td><td>開放</td></tr>
<tr><td>4</td><td>開放</td><td>－</td><td>－</td></tr>
<tr><td>5</td><td>開放</td><td>－</td><td>開放</td></tr>
<tr><td>6</td><td>開放</td><td>開放</td><td>－</td></tr>
<tr><td>7</td><td>開放</td><td>開放</td><td>開放</td></tr>
</table>

EMGOUTコネクタの詳細については、以下の説明書を参照してください。
取扱説明書コントローラセットアップと基本操作から保守まで「2.2.5 外部非常停止の接続」</td><td>7(どのエラーレベルでも開放)</td></tr>
<tr><td>CC-Link エラー解除許可
(※CR7xx-Dのみ)</td><td>E7730</td><td>整数 1</td><td>CC-Link オプション付にもかかわらず、CC-Link を接続せずにコントローラを使用した場合、エラー 7730 が発生し、操作できなくなります。このエラーは通常解除できませんが、本パラメータを使用することにより、エラーの一時解除が可能となります。
(エラー一時解除可能 / エラー解除不可 =1/0)

本パラメータは、T/B またはパソコンサポートソフトから変更した直後に有効になりますので、電源の再投入は必要ありません。逆に、本パラメータの変更内容は保存されないため、電源再投入で再び 0 に戻ります (エラー解除できなくなります)。</td><td>0(エラー解除不可)</td></tr>
</table>

<table>
<tr><th>パラメータ</th><th>パラメータ名</th><th>配列数
文字数</th><th>内容説明</th><th>出荷時設定値</th></tr>
<tr><td rowspan="4">出力信号リセットパターン

「5.14 出力信号リセットパターンについて」を参照願います。</td><td></td><td></td><td>Clr命令や専用入力(OUTRESET)などの汎用出力信号リセット時の動作を設定します。電源ON時にもここで設定したパターンで信号出力されます。
下記のパラメータで信号ごとに32ﾋﾞｯﾄ単位で設定します。
(OFF/ON/保持=0/1/*)
注)保持"*"に設定したビットは、電源ON時はOFFとなります。</td><td></td></tr>
<tr><td>ORST0</td><td>文字列 4</td><td>信号番号0～31を設定します。</td><td>00000000, 00000000,
00000000, 00000000</td></tr>
<tr><td>ORST32
：
：
：
ORST8016</td><td>文字列 4</td><td>信号番号32～63を設定します。
：
信号番号8016～8047を設定します。
注)マルチハンドをご使用の場合、出力信号716～723を使用します。
出力信号700～715はシステム予約領域のため設定はできません。</td><td>00000000, 00000000,
00000000, 00000000
：</td></tr>
<tr><td>ORS10000
：
ORS18160</td><td>文字列 4</td><td>信号番号10000～10031を設定します。
：
信号番号18160～18191を設定します。</td><td>00000000, 00000000,
00000000, 00000000
：</td></tr>
<tr><td>リセット時の出力リセット</td><td>SLRSTIO</td><td>整数 1</td><td>プログラムリセット時に汎用出力信号リセットをおこなう機能を指定します。
(無効/有効=0/1)</td><td>0(無効)</td></tr>
<tr><td>マルチ CPU 台数設定
(CR7xx-Q のみ)</td><td>QMLTCPUN</td><td>整数 1</td><td>マルチCPUシステムで、基本ベースユニットに装着するCPUユニットの台数を設定します。</td><td>2</td></tr>
<tr><td>マルチ CPUn 号機
高速通信エリア設定
(CR7xx-Q のみ)</td><td>QMLTCPUn
n=1 ～ 4</td><td>整数 4</td><td>マルチCPUシステムで、1～4号機のマルチCPU間高速通信機能の各CPUユニット間で送受信をおこなう点数を設定します。
全CPUについてパラメータ設定値を合わせる必要があります。
パラメータ設定値が合っていないとき、シーケンサCPUでエラーになりますので、各CPUのパラメータ設定値を合わせるようにしてください。

要素1:ユーザ自由エリアのサイズ(K点)
範囲:1～14(最大※)
※下表のようにマルチCPU台数によって最大値が異なります。
<table><tr><th>CPU 台数</th><th>設定範囲</th></tr><tr><td>2</td><td>0 ～ 14K 点</td></tr><tr><td>3</td><td>0 ～ 13K 点</td></tr><tr><td>4</td><td>0 ～ 12K 点</td></tr></table>
要素2:自動リフレッシュの点数(点)
範囲:0～14335
ロボットCPUは、自動リフレッシュに対応しておりませんので、自動リフレッシュの点数は、<u>常に0としてください。</u>

要素3:システムエリアのサイズ(K点)
範囲:1または2

要素4:マルチCPU同期立ち上げ(1:する、0:しない)
ロボットCPUは、立ち上がりに時間がかかりますので、基本的に1(同期する)のままで、<u>設定を変えないでください。</u></td><td>1,0,1,1</td></tr>
</table>

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
マルチ CPU 入力オフセット (CR7xx-Q のみ)	QMLTCPUS	整数 1	CR7xx-Qコントローラで、マルチCPUにおけるロボットの入力信号へのオフセットを設定します。 1Kワード単位で G10000 からのオフセットを指定し、指定した共有メモリからR/Cの入力として取り込みます。他のiQ Platform対応CPU(モーションCPUやNCCPU)を混在させて使用して、各CPUで使用する共有メモリが重複しないようにする場合、必要に応じて設定します。 設定値範囲: -1～14 (整数値) (-1:使用しない / 0～14 Kワード) (A) -1 に設定すると、装着したスロットに応じて自動的にオフセット値が固定で決められます(従来バージョン[N4a以前]と互換となります)。 (B) 0～14 に設定すると、値に応じて入力オフセットが設定されます。 ※下記「5.2.1マルチCPU入力オフセットについて(CR7xx-Qコントローラのみ)」ケース(A)、(B) を参照。 なお、ロボットを複数台接続して、このパラメータを同じ値(-1以外)にすると、シーケンサからほぼ同時に複数台のロボットに対して、同じ信号状態を入力させることもできます。 マルチCPUシステムについては、QCPUユーザーズマニュアル(マルチCPUシステム編) SH(名)-080475-F を参照してください。	-1
信号出力の処理モード 注 1)	SYNCIO	整数 1	M_Out/M_Outb/M_Out8/M_Outw/M_Out16/M_Out32/M_Dout, およびDef Ioによる信号出力時の処理モードを指定します。 従来互換モード/高速モード1/高速モード2=0/1/2 従来互換モード:信号更新周期を高速化せず、従来互換で処理します。 高速モード1: M_Out/M_Outb/M_Out8/M_Outw/M_Out16/M_Out32による信号出力を高速化します。 高速モード2: 高速モード1に加えて、M_Doutによる信号出力も高速化します。	2:高速モード2
シーケンサ入出力信号の取込設定 (CR7xx-Q のみ)	QXYREAD	整数 2	シーケンサのマルチCPUシステムで他号機CPUが管理する入出力ユニット(入出力混合ユニット含む)の入出力信号の取り込みの有効/無効を設定します。 ※信号を取り込む際に指定する信号番号は、1号機CPUのI/O割付設定に基づきます。 入出力ユニット以外の信号は更新されません。 要素1:入力信号(X)の取り込み[0:無効/1:有効] 要素2:出力信号(Y)の取り込み[0:無効/1:有効]	0,0
シーケンサ入出力ユニット設定 (CR7xx-Q のみ)	QXYUNITn n=1 ～ 4	整数 7	シーケンサのマルチCPUシステムでロボットCPUが管理CPUとなる入出力ユニット/入出力混合ユニットを設定します。 要素1:ユニットタイプ 0:対象入出力ユニット無し 1:未使用(無意) 2:出力ユニット 3:入出力混合ユニット 要素2:先頭入出力番号 0～4080(10進数) 要素3:ベース番号 0:基本ベースユニット 1～7:増設ベースユニット 要素4:スロット番号 0～11(10進数) 要素5:入出力点数 0:16点/1:32点/2:48点/3:64点/4:128点/5:256点/6:512点/7:1024点 要素6:エラー時出力モード(出力・入出力混合ユニット用) 0:クリア 1:保持 要素7:I/O応答時間1(入力・入出力混合ユニット用) 0:10ms/4:1.0ms/5:5.0ms/6:20ms/7:70ms	

注 1）本パラメータは、システム状態変数 M_Out などによる外部出力信号の処理を速くするものです。
以下に示したプログラム例 1 で、ステップ 1 および 4 の出力信号処理が高速化されます。
<プログラム例 1>

```
1 M_Out(9)=1                        ' 出力信号 9 をオンする
2 *ack_check                        '
3 If M_In(7)=0 Then *ack_check      ' 入力信号 7 がオンするまで待つ ( インターロック )
4 M_Out(9)=0                        ' 出力信号 9 をオフする
5 End
```

＊高速化の参考値：上記プログラム例では処理時間がおよそ 80% 削減されます。
但し、CR7xx-D コントローラの CC-Link、プロフィバス、パラレル入出力インタフェース (カード) では、外部信号出力の命令が複数行連続している場合に効果があります。下記プログラム例 2 では処理時間がおよそ 75% 削減されます。
<プログラム例 2>

```
1 M_Out(9)=1                    ' 出力信号 9 をオンする
2 M_Out8(10)=&H1F               ' 出力信号 10 から 8 ﾋﾞｯﾄに &H1F を出力する
3 M_Out16(18)=&H3FFF            ' 出力信号 18 から 16 ﾋﾞｯﾄに &H3FFF を出力する
4 M_Out32(33)=&H7FFFFFFF        ' 出力信号 33 から 32 ﾋﾞｯﾄに &H7FFFFFFF を出力する
5 End
```

⚠注意　ロボットと周辺装置間で信号を確実に受け渡すために、信号のインタロックは必ず実施するようにしてください。インターロックをおこなわないと信号が正しく伝わらず誤動作の原因となります。

5.2.1 マルチCPU入力オフセットについて(CR7xx-Qコントローラのみ)

(1) ケース (A)

入力にオフセットを使用しない場合 (パラメータ : QMLTCPUS = -1 の時)

表 5-3 : CPU 共有メモリとロボット入出力信号の対応

シーケンサ (ワードデバイス)		ロボット (ビットデバイス)	
出力	U3E0¥G10000 ～ U3E0¥G10511	入力	ロボット CPU No.1 / 10000 ～ 18191
	U3E0¥G10512 ～ U3E0¥G11023		ロボット CPU No.2 / 10000 ～ 18191
	U3E0¥G11024 ～ U3E0¥G11535		ロボット CPU No.3 / 10000 ～ 18191
入力	U3E1¥G10000 ～ U3E1¥G10511	出力	ロボット CPU No.1 / 10000 ～ 18191
	U3E2¥G10000 ～ U3E2¥G10511		ロボット CPU No.2 / 10000 ～ 18191
	U3E3¥G10000 ～ U3E3¥G10511		ロボット CPU No.3 / 10000 ～ 18191

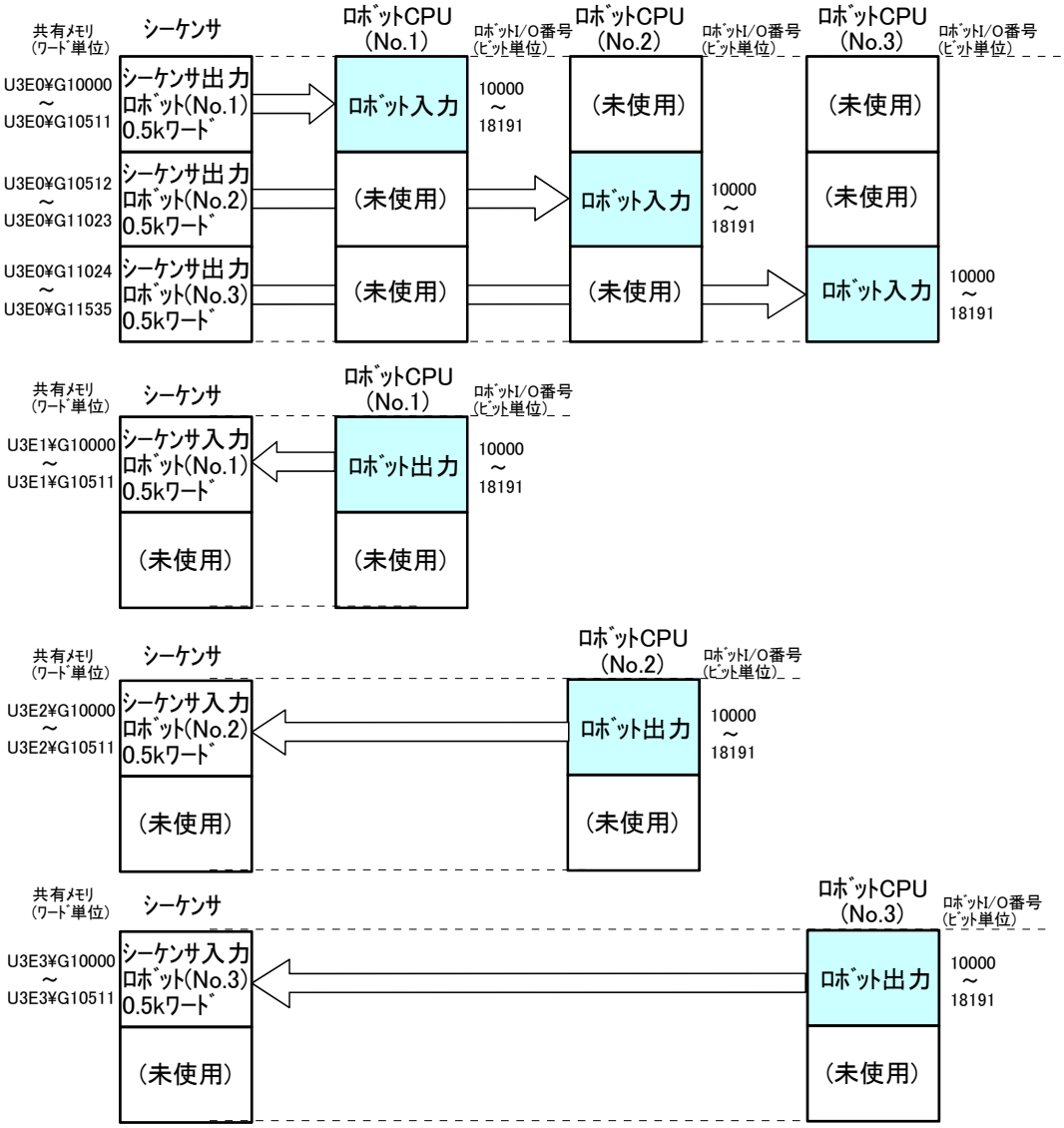


図 5-1 : CPU 共有メモリとロボット入出力信号の対応 (ケース (A))

(2) ケース (B)

入力にオフセットを使用する場合 (パラメータ QMLTCPUS = 0 ～ 14 の時)

$\alpha(1)$ = (ロボット CPU No.1 の QMLTCPUS) * 1024

$\alpha(2)$ = (ロボット CPU No.2 の QMLTCPUS) * 1024

$\alpha(3)$ = (ロボット CPU No.3 の QMLTCPUS) * 1024

表 5-4 : CPU 共有メモリとロボット入出力信号の対応

シーケンサ (ワードデバイス)		ロボット (ビットデバイス)	
出力	U3E0¥G10000+$\alpha(1)$ ～ U3E0¥G10511+$\alpha(1)$	入力	ロボット CPU No.1 / 10000 ～ 18191
	U3E0¥G10000+$\alpha(2)$ ～ U3E0¥G10511+$\alpha(2)$		ロボット CPU No.2 / 10000 ～ 18191
	U3E0¥G10000+$\alpha(3)$ ～ U3E0¥G10511+$\alpha(3)$		ロボット CPU No.3 / 10000 ～ 18191
入力	U3E1¥G10000 ～ U3E1¥G10511	出力	ロボット CPU No.1 / 10000 ～ 18191
	U3E2¥G10000 ～ U3E2¥G10511		ロボット CPU No.2 / 10000 ～ 18191
	U3E3¥G10000 ～ U3E3¥G10511		ロボット CPU No.3 / 10000 ～ 18191

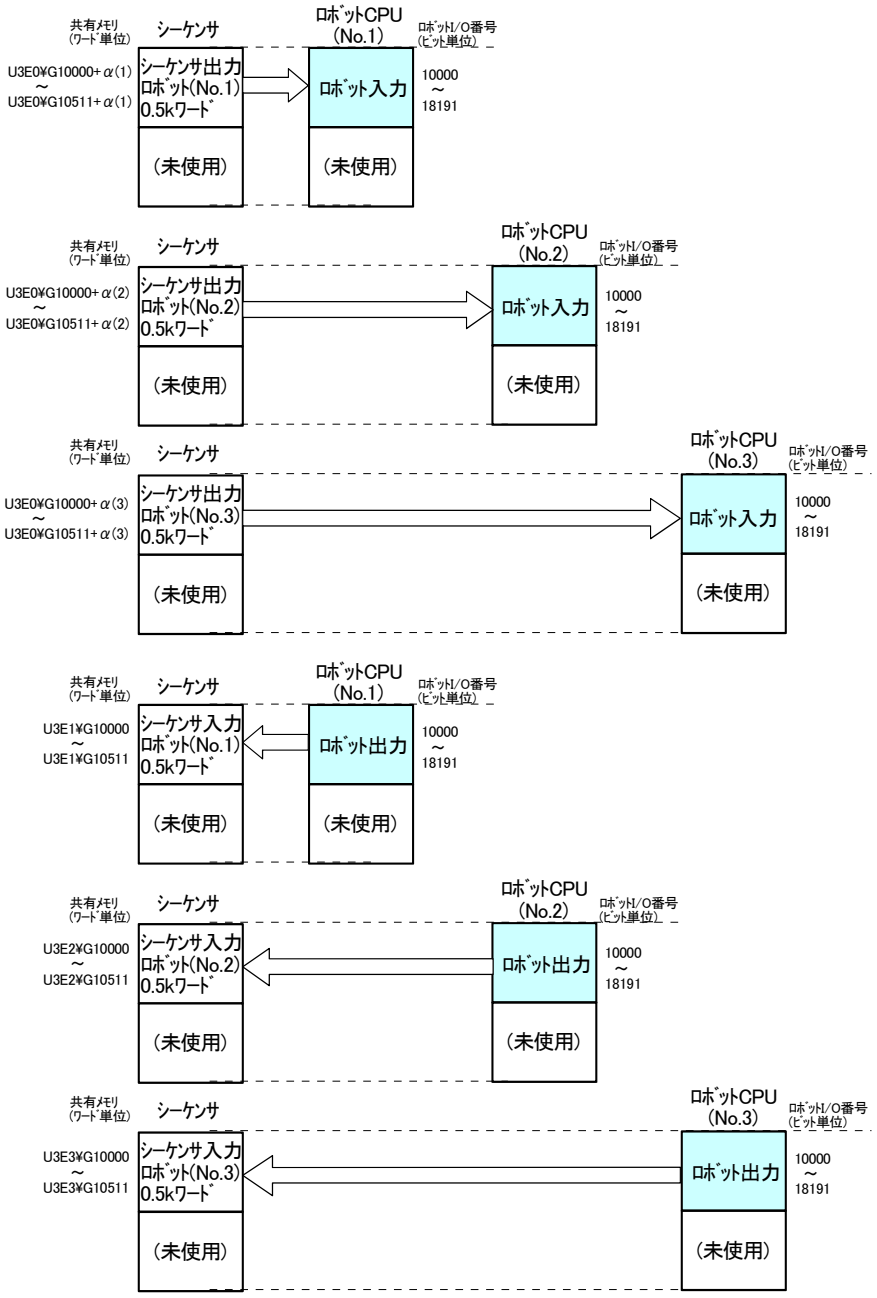


図 5-2 : CPU 共有メモリとロボット入出力信号の対応 (ケース (B))

5.3 操作パラメータ

コントローラ、T/B 等の操作に関するパラメータです。

表 5-5：操作関連パラメーター覧

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ブザーの ON/OFF	BZR	整数 1	コントローラにてエラー発生時などになるブザー音のON/OFFを指定します。 (OFF/ON=0/1)	1(ON)
プログラムリセット操作権	PRSTENA	整数 1	プログラムリセット操作の操作権が必要か不要か。 (必要/不要=0/1) 操作権不要にした場合、どこからでもプログラムリセットが可能になります。ただし、ティーチモードでは安全上できません。	0(必要)
モード切替え時プログラムリセット	MDRST	整数 1	コントローラのモード切替え時に, プログラム中断状態を解除します。 (無効/有効=0/1)	0(無効)
操作パネル表示モード	OPDISP	整数 1	コントローラのモードを切替えたときの5桁LED表示モードの設定。 0:モード変更時にオーバーライド表示になります。(初期値) 1:モードを変更しても現状の表示モードを保持します。	0(オーバーライド表示)
プログラム選択権設定	OPPSL	整数 1	操作パネルに操作権があるとき、外部信号からのプログラム選択操作も有効にします。 (外部/OP=0/1)	1(OP)
	RMTPSL	整数 1	外部入出力に操作権があるとき、操作パネルからのプログラム選択操作も有効にします。 (外部/OP=0/1)	0(外部)
T/B のオーバーライド操作権	OVRDTB	整数 1	T/Bからのオーバーライド変更時に操作権が必要かを指定します。 (不要/必要=0/1) 注)OVRDTB=1(操作権必要)に設定した場合、OVRDENAの設定とコントローラのモードにより、T/Bの操作権の要(T/B有効時)/不要(T/B無効時)が変わります。下表を参照してください。	1(必要)
モード変更時の速度設定	OVRDMD	整数 2	モード変更時に自動的にオーバーライドを設定します。 第1要素 … "MANUAL"から"AUTOMATIC"に変更された場合のオーバーライド値 第2要素 … "AUTOMATIC"から"MANUAL"に変更された場合のオーバーライド値 0に設定した場合は現状維持になります。	0,0
オーバーライド変更操作権	OVRDENA	整数 1	操作パネルおよび外部からのオーバーライド変更時に操作権が必要かを設定をします。 (必要/不要=0/1) 注)OVRDENA=1(操作権不要)に設定してあっても、コントローラのモードが"MANUAL"のときはオーバーライドの変更はできません。	0(必要)
プログラムのアクセス対象切り替え (ROM/RAM 運転)	ROMDRV	整数 1	プログラムのアクセス対象を ROM ⇔ RAM に切り替えます。 0=RAM モード (初期値 SRAM メモリ使用) 2= 高速 RAM モード (DRAM メモリ使用) 注) 値 "1" は未使用。 高速 RAM モードとコンティニュー機能は併用できません。高速 RAM モードを使用する場合は、コンティニュー機能 (パラメータ CTN) を無効にしてください。	2(DRAM)
RAM 領域の情報を ROM 領域へコピー	BACKUP	文字列 1	プログラム、パラメータ、共通変数、エラーログを RAM 領域から ROM 領域へコピーします。 本パラメータの値は変更しないでください。	SRAM->FLROM (固定)
ROM 領域の情報を RAM 領域へ書き戻す	RESTORE	文字列 1	ROM 領域のプログラム、パラメータ、共通変数、エラーログを RAM 領域へ書き戻します。 本パラメータの値は変更しないでください。	FLROM->SRAM (固定)

OVRDTB のオーバーライド操作権 の内容説明 内の表：

コントローラのモード	OVRDENA=0	OVRDENA=1
AUTOMATIC	変更不可	変更可 (TB 無効時)
MANUAL	変更可 (TB 有効時)	変更可 (TB 有効時)

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
メンテ予報の 実行間隔	MFINTVL	整数 2	メンテ予報用のデータの収集間隔を設定します。 第1要素 … データの収集レベル 1(低)～5(高) 第2要素 … 予報チェックの実行間隔(単位:時間)	1(低)、6(時間)
メンテ予報の お知らせ方法	MFREPO	整数 2	メンテ予報のお知らせ方法を設定します。警告や信号出力を止めたいときは 0 に設定してください。 第1要素…警告を発生させる/させない(1/0) 第2要素…専用信号の出力する/しない(1/0)	1(警告を発生)、 0(信号出力しない)
メンテ予報の リセット 注) 本パラメータを T/B から読み出すときは、すべてのパラメータ名を入れて読み出してください。	MFGRST	整数 1	メンテ予報機能のグリースに関する累積データをリセットします。 ※ メンテ予報機能のグリースの補給を促す警告 (7530 番台) が発生し、グリース補給を実施した軸は、コントローラに累積されたグリースに関するデータをリセットする必要があります。 基本的には、パソコンサポートソフトウェアのメンテ予報画面からリセット操作をおこないますが、パソコンが準備できない場合、ティーチングボックスから本パラメータを入力することにより、累積されたデータをリセットすることができます。	0:全軸リセット 1～8:指定軸単位にリセット
	MFBRST	整数 1	メンテ予報機能のベルトに関する累積データをリセットします。 ※ メンテ予報機能のベルトの交換を促す警告 (7540 番台) が発生し、ベルト交換を実施した軸は、コントローラに累積されたベルトに関するデータ をリセットする必要があります。 基本的には、パソコンサポートソフトウェアのメンテ予報画面からリセット操作をおこないますが、パソコンが準備できない場合、ティーチングボックスから本パラメータを入力することにより、累積されたデータをリセットすることができます。	0:全軸リセット 1～8:指定軸単位にリセット
位置復旧支援	DJNT	実数 8	位置復旧支援ツールによって求められた原点の補正データが入ります。位置復旧支援ツール以外では変更しないでください。パソコンサポートソフトウエアのパラメータ専用画面にて参照のみできます。	機種によって異なります。
	MEXDTL	実数 6	位置復旧支援ツールによって求められた標準ツールの補正データが入ります。位置復旧支援ツール以外では変更しないでください。	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
	MEXDTL1	実数 6	位置復旧支援ツールによって求められたツール番号 1 用の補正データが入ります。位置復旧支援ツール以外では変更しないでください。	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
	MEXDTL2	実数 6	位置復旧支援ツールによって求められたツール番号 2 用の補正データが入ります。位置復旧支援ツール以外では変更しないでください。	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
	MEXDTL3	実数 6	位置復旧支援ツールによって求められたツール番号 3 用の補正データが入ります。位置復旧支援ツール以外では変更しないでください。	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
	MEXDTL4	実数 6	位置復旧支援ツールによって求められたツール番号 4 用の補正データが入ります。位置復旧支援ツール以外では変更しないでください。	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
	MEXDBS	実数 6	位置復旧支援ツールによって求められたベース用の補正データが入ります。位置復旧支援ツール以外では変更しないでください。	(X,Y,Z,A,B,C) = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
T/B からの起動画面 有効 / 無効	TBOP	整数 1	T/Bからのプログラム起動の使用可能/不可能を設定します。 0:仕様不可/1:使用可能	1
2 重系エラーに関する [RESET] キー入力時 の挙動選択 ※ 以下の S/W Ver でのみ有効。 S/W Ver. F-Q ｼﾘｰｽﾞ : R6b 以降 F-D ｼﾘｰｽﾞ : S6b 以降	CATEGORY	整数 1	2重系エラー発生時における、[RESET]キーによる挙動を設定します。 3:エラーリセット可 4:エラーリセット不可 対象となるエラーは次のとおりです。 H0039(ドアスイッチ開放信号配線異) H0046(イネーブリングデバイス配線異) H0051(外部非常停止配線異常)	3:エラーリセット可

5.4 プログラムパラメータ

プログラム実行及びロボット言語に関するパラメータです。

表 5-6：プログラム実行関連パラメーター覧

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
マルチタスク本数	TASKMAX	整数 1	同時に実行するプログラムの数を指定します。	8
スロット表 （マルチタスク運転時に設定）			マルチタスク運転時に各タスクスロットの運転条件を設定します。プログラムリセット時に、設定されます。	
	SLT* ＊は 1 ～ 32	文字列 4	「プログラム名」,「運転モード」,「起動条件」,「優先順位」を指定します。 プログラム名：選択されるプログラム名。アルファベットを使用する場合は大文字を使用してください。小文字は認識できません。 運転モード：連続/1サイクル=REP/CYC REP：プログラムが繰り返し実行されます。 CYC：1サイクル実行して終了します。(プログラムがGoto命令などにより無限ループになっている場合は終了しません。) 起動条件：通常/エラー時/常時=START/ERROR/ALWAYS START：操作パネルのSTARTボタンや信号の始動信号により実行されます。 ALWAYS：コントローラの電源ON後即実行されます。このプログラムは始動中などの状態には影響しません.。 ALWAYS属性に設定してあるプログラムを編集するには一度常時実行の属性を解除してから編集してください。 常時実行属性のプログラムは常に実行されていますので編集ができません。ALWAYSをSTARTに変更しコントローラの電源を再投入して常時実行を停止させてください。 ERROR: エラー発生時に実行されます。このプログラムは始動中などの状態には影響しません。 起動条件にてALWAYS、ERRORを設定したプログラムでは下記の動作系の命令は実行できません。実行時にエラーになります。 Mov, Mvs, Mvr, Mvr2, Mvr3, Mvc, Mva, Drive, Getm, Relm, JRC 優先順位：1～31(31が最大) この数値は一度に実行する行数を表します。Priority命令の実行行数と同じ意味です。 たとえば2つのスロットを使用して実行する場合にSLT1を1、SLT2を2と指定した場合はSLT1のプログラムを1行実行後SLT2のプログラムを2行実行します。 よってSLT2のプログラムの方が多く実行され、結果的にSLT2の方が優先順位が上がることになります。	””,REP,START,1
プログラム選択記憶	SLOTON	整数 1	プログラム選択時にプログラム名をパラメータに記憶するかどうか、また、サイクル運転終了時にプログラム選択状態を保持するかどうかを指定します。 (1)プログラム選択時のプログラム名記憶有効 （ビット0、記憶有効/記憶無効=1/0） 記憶有効：スロット1のプログラム選択時に、SLT1パラメータに現在のプログラム名を保存します。また、電源ON時にSLT1パラメータに設定されているプログラムを選択します。 記憶無効：スロット1のプログラム選択時に、SLT1パラメータに現在のプログラム名を保存しません。電源ON時は、記憶有効時と同じく、にSLT1パラメータに設定されているプログラムを選択します。 (2)サイクル運転終了時プログラム保持 （ビット1、保持/非保持=1/0） 保持　：サイクル運転終了時にプログラム選択状態を保持します。P.0000になりません。 非保持：サイクル運転終了時にプログラム選択状態を保持しません。P.0000になります。 設定値と動作 0：記憶無効、非保持 1：記憶有効、非保持(初期値) 2：記憶無効、保持 3：記憶有効、保持	1(記憶有効、非保持)

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
常時実行プログラム内での X** 命令、SERVO 命令、RESET ERR 命令を実行可能にする設定	ALWENA	整数 1	SLT*パラメータにて常時実行に設定したプログラム(起動条件がALWAYS)内にてXrun、Xload、Xstp、Xrst、Servo、Reset Error命令を実行可能にします。 可/不可=1(0以外)/0	0(不可)
ユーザベースプログラム 「4.4.24 ユーザ定義外部変数」を参照願います。	PRGUSR	文字列 1	ユーザベースプログラム名を設定します。アルファベットを使用する場合は大文字を使用してください。小文字は認識できません。 ユーザベースプログラムはユーザ定義外部変数を使用したい場合に設定するプログラムです。中身はDef InteやDimなどの変数定義命令のみを記述します。 ユーザベースプログラムでDim命令にて配列変数を宣言した場合は使用するプログラムにて再度Dim命令で同じ変数名を定義する必要があります。配列でない変数の場合は再定義は不要です。	""(なし)
コンティニュー機能	CTN	整数 1	プログラム実行スロット1についてのみ、電源OFF時の状態を保存して、次回の電源ON時に保存状態から継続して動作させることができます。保存されるデータは、プログラム実行環境(オーバライド、実行ステップ行、プログラム内変数等)と出力信号状態です。本機能が有効の時、運転中に電源OFFすると、次回の電源ON時に待機中の状態で立ち上がります。継続する場合は、サーボ電源を投入して始動入力してください。 (機能無効/有効=0/1) <注意事項> 1)ブレーキのついていない軸があるロボットについては電源を切った場合にアームが自重で下がったり、回転してしまう場合がありますので、本機能をご使用される場合は十分注意する必要があります。 2)コンティニュー機能で継続できるプログラムはタスクスロット1にロードされているもののみです。スロット2以降は継続実行されず、プログラムリセット状態で立ち上がります。 3)本機能を有効にした後は下記パラメータは変更できなくなりますので本パラメータを変更する前に変更しておいてください。 SLTn、SLOTON、TASKMAX。 4)本機能を有効にした後にスロット表(SLT*)パラメータを変更しても、スロット表へ反映されません。一旦コンティニュー機能を無効にした後、電源をOFF, ONしてスロット表(SLT*)パラメータを変更してください。 5)高速RAMモードとコンティニュー機能は併用できません。コンティニュー機能を使用する場合は、高速RAMモード(パラメータROMDRV)を無効にしてください。	0(無効)
JRC 命令 (軸の多回転機能)			JRC命令の実行状態を設定します。	
	JRCEXE	整数 1	JRC命令の実行可/否を指定します。 実行可/不可=(1/0)	0(実行不可)
	JRCQTT	実数 8	JRC命令で増減させる変化量を、先頭要素からJ1,J2,J3,の順でJ8軸まで設定します。ユーザ定義軸のみで有効です。 ロボット付加軸ではJ7, J8軸、メカ付加軸では任意軸の設定が有効となります。単位はパラメータAXUNTに依存します。	JRC実行可能 ロボット 0,0,0,0,0,360,0,0 または 0,0,0,360,0,0,0,0 JRC実行不可 ロボット 0,0,0,0,0,0,0,0
	JRCORG	実数 8	JRC 0命令を実行して原点設定する時の原点座標値を設定します。 ユーザ定義軸のみで有効です。 単位はパラメータAXUNTに依存します。	0,0,0,0,0,0,0,0
付加軸設定	AXUNT	整数 16	付加軸の単位系を設定します。 角度(degree)/長さ(mm) = 0/1	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
ユーザエラー設定	UER1 ～ UER20	整数 1、 文字列 3	Error命令によるエラーのメッセージ、原因、復旧方法を設定します。 最大20のユーザエラーについて設定が可能です。 第1要素…設定をおこなうエラー番号(9000～9299が有効) 初期値の9900は使用できません。変更してご利用ください 第2要素…エラーメッセージ 第3要素…原因 第4要素…復旧方法 メッセージ中にスペースを含む場合は全体を" " "で囲ってください 例) 9000, "Time Out", "No Signal", "Check Button"	9900,"message","cause","treat"

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
位置データの回転成分の単位設定	PRGMDEG	整数 1	ロボットプログラム内で位置データの回転成分の単位を指定します。 0:RAD 1:DEG 例)M1=P1.A（このときの単位を指定します。） (デフォルトでは成分データを参照するときはラジアンです。) 位置定数(P1=(100,0,300,180,0,180)(7,0))の回転成分はデフォルトでDEGです。このパラメータとは関係ありません。	0(RAD)
ロボット言語	RLNG	整数 1	使用するロボット言語を選択します。 2：MELFA-BASIC V 1：MELFA-BASIC IV	2
表示言語[注 1)]	LNG	文字列 1	表示言語を設定します。 ″JPN″：日本語表示 ″ENG″：英語表示 以下に示す表示言語が変更されます。 (1)パソコンサポートソフトウェア ・ロボットのアラームメッセージ ・パラメータ説明リスト (2)外部通信でロボットから読み出したアラームメッセージ（ロボットCPUのイーサネットインタフェース） ※R33TB、または R57TB をご使用の場合は、それぞれ専用の画面で設定します。(R33TB は本書 95 ページの「3.18 初期設定画面の操作」を、R57TB は別冊の取扱説明書「R57TB 取扱説明書」を参照願います）	日本語仕様の場合は、″JPN″ 英語仕様の場合は、″ENG″
外部変数拡張	PRGGBL	−	本パラメータを 0 にし電源を再投入するとプログラム外部変数の容量が半分になりますが、プログラム保存領域が 920KB から 940KB に増えます。	1
コンパイルモード	PRGMODE	整数 1	コンパイル実行モードを有効にすることにより、プログラム処理速度が向上します。 0：コンパイル無効 1：コンパイル有効	1

注 1）別の言語の取扱説明書（日本語または英語）をご希望の場合はご用命願います。
また、ロボット本体とコントローラに貼り付けてある注意シールは、ご注文仕様に基づいた言語で製作しております。本パラメータを変更して言語を切り替えてご使用の場合はご注意願います。

5.5 通信パラメータ

通信に関するパラメータです。

表 5-7 : 通信関連パラメーター覧

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
RT Tool Box 2 通信方式設定	COMSPEC	整数 1	コントローラとRT Tool Box 2との通信方式(従来通信方式/高信頼通信方式)を指定します。 0:従来通信方式 1:高信頼通信方式 従来通信方式に比べ、速度は遅くなりますが、ノイズの多い周辺環境においても信頼性の高い通信を実現します。	1
通信設定	COMDEV	文字列 8	MELFA-BASIC VのOpen命令に通信回線を使用する場合のCOM1:COM2:にどの回線を割り当てるかの設定をおこないます。このパラメータは、データリンク(Open命令で使用する)をおこなう場合に設定が必要です。 プログラムのOpen文で指定するCOMn:に対応するデバイスをこのパラメータで指定します(nは1～8)。 パラメータ要素の左から順にCOM1:、COM2:.....COM8:の順です。 データリンクをおこなう場合に設定が必要です。 OPT11～OPT19を割付けします。(下記、イーサネット関係のパラメータを参照)	, , , , , , ,
イーサネット	NETIP	文字列 1	コントローラのIPアドレス。 F-Dシリーズコントローラをご使用の場合、192.168.100.xxxのアドレスを設定することはできません。 RT ToolBox2でパラメータをリストアした場合、電源リセット2回でNETIPが有効となります。	F-Dシリーズ: 192.168.0.20 F-Qシリーズ: 192.168.100.1
	NETMSK	文字列 1	サブネットマスク	255.255.255.0
	NETPORT	数値 10	ポート番号 範囲 0～32767 リアルタイム外部制御用, ---------- 10000 , COMDEV のOPT11～19に対応 (OPT11) 10001 , (OPT12) 10002 , (OPT13) 10003 , (OPT14) 10004 , (OPT15) 10005 , (OPT16) 10006 , (OPT17) 10007 , (OPT18) 10008 , (OPT19) 10009	
	CPRCE11 CPRCE12 CPRCE13 CPRCE14 CPRCE15 CPRCE16 CPRCE17 CPRCE18 CPRCE19	数値 9	プロトコル 0:無手順, 1:有手順, 2:データリンク (1:有手順は現在機能がありません) COMDEV のOPT11～19に対応 (OPT11) 0 , (OPT12) 0 , (OPT13) 0 , (OPT14) 0 , (OPT15) 0 , (OPT16) 0 , (OPT17) 0 , (OPT18) 0 , (OPT19) 0	

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
イーサネット	NETMODE	数値 9	サーバー指定(1:サーバー、0:クライアント) (OPT11) (OPT12) (OPT13) (OPT14) (OPT15) (OPT16) (OPT17) (OPT18) (OPT19)	 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1
	NETHSTIP	文字列 9	データ通信先のサーバーのIPアドレス ※ NETMODEでクライアント指定したときのみ有効 (OPT11) (OPT12) (OPT13) (OPT14) (OPT15) (OPT16) (OPT17) (OPT18) (OPT19)	 192.168.0.2 , 192.168.0.3 , 192.168.0.4 , 192.168.0.5 , 192.168.0.6 , 192.168.0.7 , 192.168.0.8 , 192.168.0.9 , 192.168.0.10
	MXTTOUT	数値 1	リアルタイム外部制御命令実行時のタイムアウト時間 (7.1msecの倍数、-1でタイムアウトなし)	-1
ジョブロードタイムアウト時間	NVJBTOUT	整数 1	ネットワークビジョンのジョブロードタイムアウト時間を設定します。 (単位:秒、設定範囲:1～32767)	90

5.6 標準ツール座標について

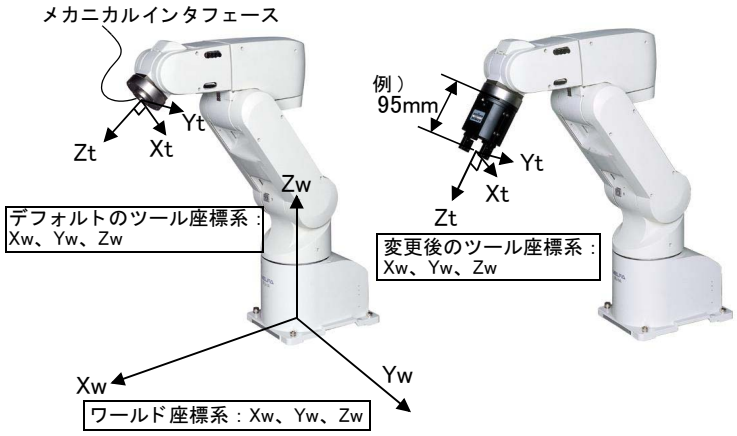
ロボットにハンドを取り付けた場合にロボットの制御点をハンド先端にする場合にツールデータを設定する必要があります。設定方法は下記の 3 点です。

1) パラメータ MEXTL にて設定する。
2) ロボットプログラム内で Tool 命令にて設定する。
3)M_Tool 変数へツール番号をセットする。MEXTL1 ～ 16 パラメータで設定されている値がツールデータになります。
詳細は 339 ページの「M_Tool」をご覧ください。

出荷時設定はゼロで設定されており、制御点はメカニカルインタフェース (フランジ面) です。

ツールデータの構造：X，Y，Z，A，B，C
X，Y，Z 軸：メカニカルインタフェースからのツール座標系での値
A 軸　　：ツール座標系の X 軸回転
B 軸　　：ツール座標系の Y 軸回転
C 軸　　：ツール座標系の Z 軸回転

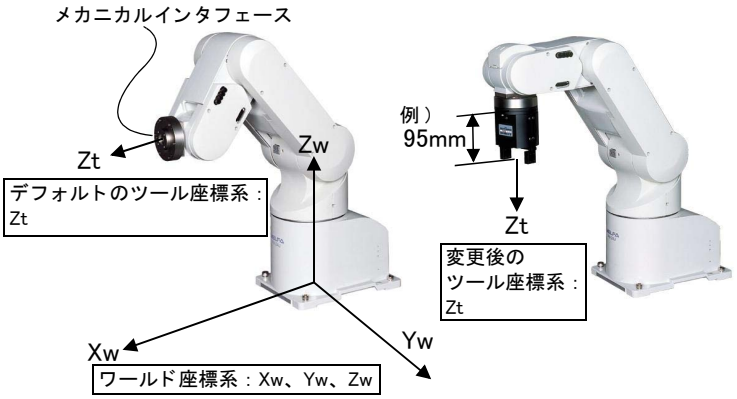
垂直 6 軸ロボットの例



＜垂直 6 軸ロボットの例＞
1) パラメータでの設定例
　パラメータ名：MEXTL
　値：0，0，95，0，0，0
2)Tool 命令の設定例
　1 Tool (0,0,95,0,0,0)

6 軸ロボットは、動作範囲内であればいろいろな姿勢が可能です。

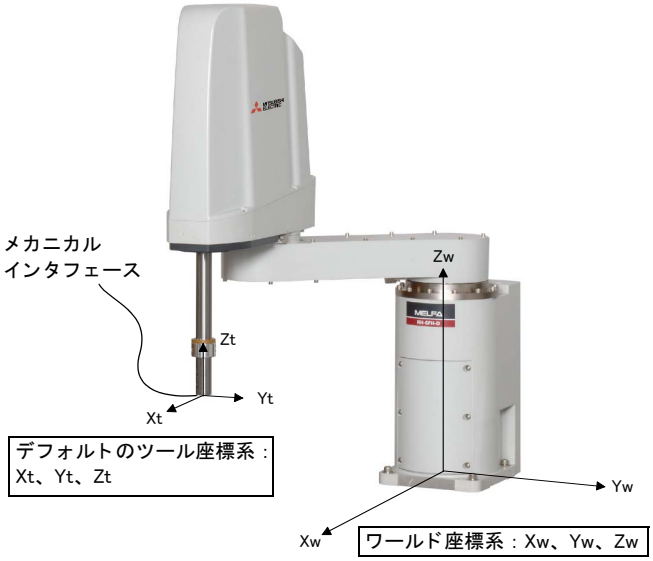
垂直 5 軸ロボットの例



＜垂直 5 軸ロボットの例＞
1) パラメータでの設定例
　パラメータ名：MEXTL
　値：0，0，95，0，0，0
2)Tool 命令の設定例
　1 Tool (0,0,95,0,0,0)

5 軸ロボットは動作範囲の関係で Z 軸成分のみ有効です。他の軸にデータを入れても無視されます。

水平 4 軸ロボットの例



＜水平 4 軸ロボットの例＞
1) パラメータでの設定例
　　パラメータ名：MEXTL
　　値：0，0，−10，0，0，0
2)Tool 命令の設定例
　　1 Tool (0,0,−10,0,0,0)

水平 4 軸ロボットは基本的には平行移動でのオフセットが可能です。
垂直ロボットとはツール座標系の向きが異なりますのでご注意ください。

ツール変換データの軸成分は、ロボットの機種によって有効なものとそうでないものがあります。
表 5-8 を参照して、適切なデータを設定してください。

表 5-8：ロボットの形式に対するツール変換データの有効な軸成分

形式	軸数	ツール変換データの軸成分[注 1)]					
		X	Y	Z	A	B	C
RV-2F、RV-4F/4FL、RV-7F シリーズ、RV-13F シリーズ、RV-20F、RV-50F シリーズ	6	○	○	○	○	○	○
RV-4FJL	5	△	△	○	△	△	×
RH-3FH/6FH/12FH/20FH シリーズ、RH-3FHR シリーズ	4	○	○	○	△	△	○

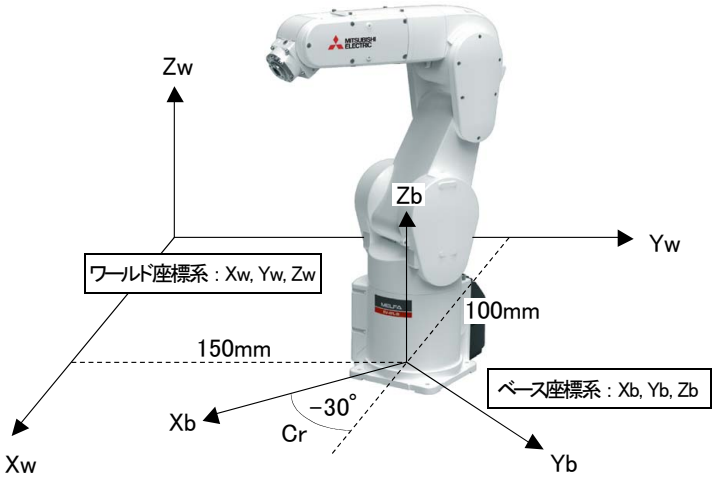
注 1）○：有効。△：未使用、設定しても無視されます。×：設定値は必ず 0 に固定してください、0 以外を設定すると動作に支障がでる場合があります。

5.7 標準ベース座標について

出荷時にワールド座標系位置は 0(ｾﾞﾛ)に設定されており、ベース座標系(ロボットの据付位置)はワールド座標系(ロボットの現在位置の基準となる座標系)に一致しています。
ベース変換機能を利用することにより、ワールド座標系原点を J1 軸の中心位置以外に設定することができます。
ベース変換を実行すると、ワールド座標系とベース座標系の位置関係が変更されるため、それまでの教示位置へロボットを移動させると以前とは異なった位置へ移動しますので、教示位置とベース変換の関係を確実に管理して、ベース変換機能を有効に活用してください。設定方法は以下の 4 種類です。

1) パラメータ：MEXBS へのベース変換データの直接設定
2) パラメータ：MEXBSNO へのベース座標番号による設定
3) パラメータ：J1OFFSET による J1 軸オフセット角度の設定(垂直 5 軸タイプのロボットのみ)
4) ロボットプログラムでの Base 命令の実行による設定

ベース座標系データの構造：X，Y，Z，A，B，C
X，Y，Z 軸：ワールド座標原点からのベース座標系(ロボットの据付位置)の位置
A 軸　　：ワールド座標系の X 軸回転
B 軸　　：ワールド座標系の Y 軸回転
C 軸　　：ワールド座標系の Z 軸回転



注）図は垂直 6 軸ロボットの例です。水平多関節型ロボットでも同じです。

(例)
1) パラメータでの設定例
　パラメータ名：MEXBS
　値：100,150,0,0,0,−30
2)Base 命令の設定例
　1 Base (100,150,0,0,0,−30)

通常はベース座標系は変更しなくても問題ありませんが、変更する場合は上記例を参考の上設定してください。ロボットプログラム内での Base 命令での変更は思わぬ位置へ移動してしまう場合がありますので十分注意して実行してください。

ベース変換データの軸成分は、ロボットの機種によって有効なものとそうでないものがあります。
表 5-9 を参照して、適切なデータを設定してください。

表 5-9：ロボットの形式に対するベース変換データの有効な軸成分

形式	軸数	ベース変換データの軸成分[注 1)]					
		X	Y	Z	A	B	C
RV-2F、RV-4F/4FL、RV-7F シリーズ、RV-13F シリーズ、RV-20F、RV-50F シリーズ	6	○	○	○	○	○	○
RV-4FJL シリーズ	5	○	○	○	△	△	○[注 2)]
RH-3FH/6FH/12FH/20FH シリーズ、RH-3FHR シリーズ	4	○	○	○	△	△	○

注 1）○：有効、△：未使用、×：設定値は 0 に固定してください。
注 2）パラメータ：J1OFFSET へ値を設定してください。

⚠注意

ベース変換をおこなうとロボットの現在位置基準が変わるため、それまでの教示データはそのままでは使えなくなります。
誤ってベース変換前に教示した位置へ動作させると、ロボットは思わぬ位置へ移動し物損や人身事故の原因となります。
ベース変換機能をご使用になる場合は、変換するベース座標系と教示位置の関係をしっかりと管理して、ロボットを正しく動作させるよう注意してベース変換機能をご活用いただきますようお願いします。

5.8 ユーザ定義領域について

ユーザ定義領域は、パラメータで設定する任意の位置領域にロボット制御点が入っているか否かを常時監視する機能です。専用入出力や状態変数を用いて、領域内 / 外の状態を出力する機能と、領域内時にエラー停止する機能の中から選択することができ、周辺装置と作業空間を共有する場合の周辺装置との連携や、周辺装置との干渉回避などに有効です。また、位置領域以外にも、ロボットの姿勢領域判定や、付加軸領域判定もご利用できます。

本機能は以下のパラメータ設定で使用可能となります。

1) 基準とする座標系の選択 (パラメータ : AREAnCS)
2) ユーザ定義領域の設定 (パラメータ : AREAnP1、AREAnP2)
3) 対象メカの指定 (パラメータ : AREAnME)
4) ユーザ定義領域内での挙動指定 (パラメータ : AREAnAT)

以下に各パラメータの設定内容詳細を説明します。

5.8.1 座標系の選択

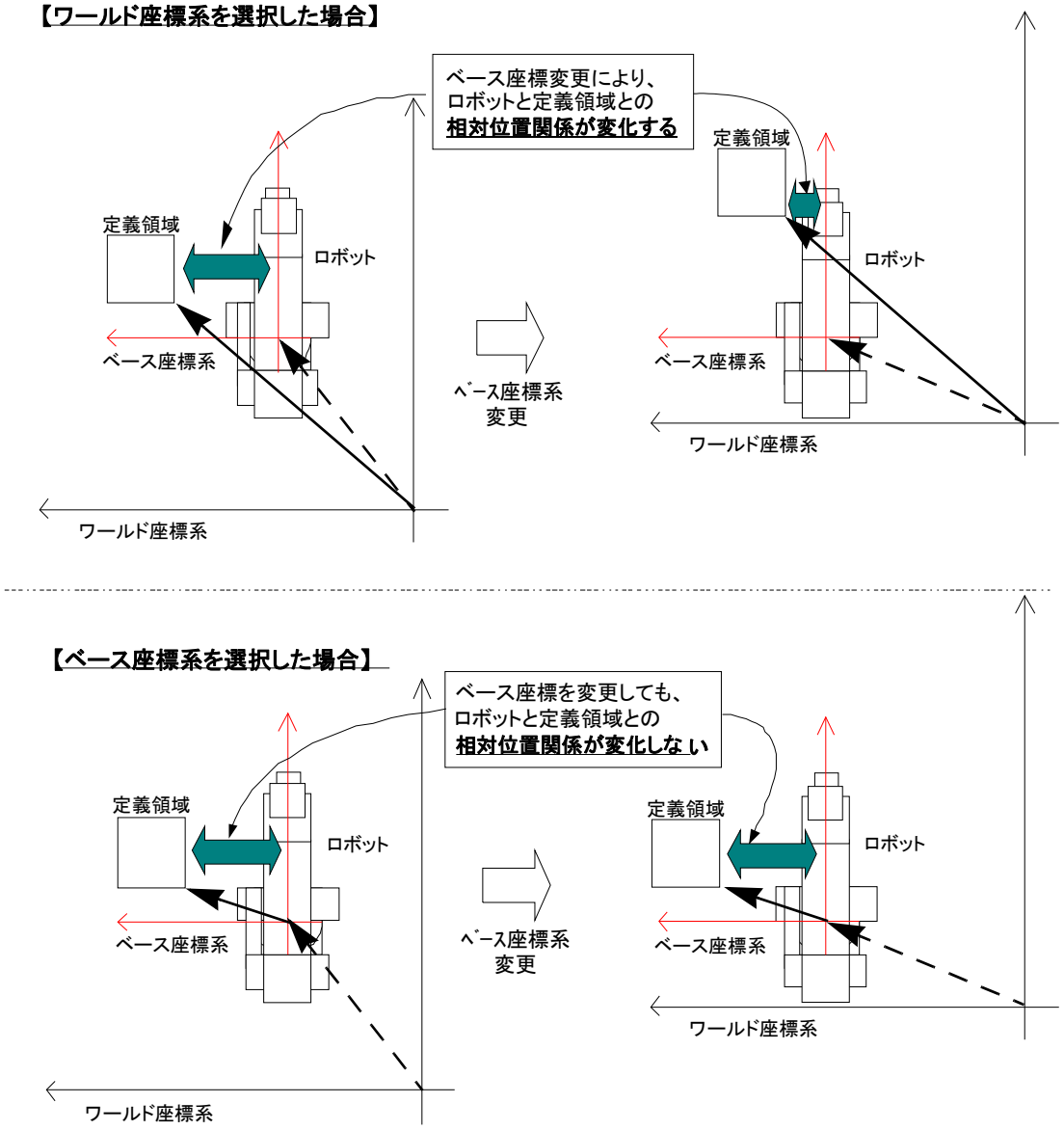
この機能は、Base 命令等を用いてベース座標系を変更してご使用になられる場合、ユーザ定義領域も同時に移動させるか、固定しておくかを選択するものです。選択はパラメータ：AREAnCS にて、基準とする座標系を「ワールド座標系」（移動させる）、または「ベース座標系」（固定しておく）のいずれかを指定します。

ベース座標系を変更されない場合は、どちらを選択してもユーザ定義領域は変化しません。

座標系の詳細については、「4.5 ロボットの座標系解説」を参照願います。

表 5-10：基準とする座標系

座標系	説明（特徴）
ワールド座標系	ベース座標系を変更すると、ユーザ定義領域も同時に移動します。ロボット本体とユーザ定義領域の相対位置関係が変わります。
ベース座標系	ベース座標系を変更しても、ユーザ定義領域は移動しません。ロボット本体とユーザ定義領域の相対位置関係は固定です。 ベース座標系を変更して使用するが、ロボット本体とユーザ定義領域の相対位置関係を固定させておきたい場合に有効です。



注）図は垂直 6 軸ロボットを例に真上から見たイメージを示します。水平多関節型ロボットでも同じです。

図 5-3：ワールド座標系とベース座標系での違い

5.8.2 領域の定義

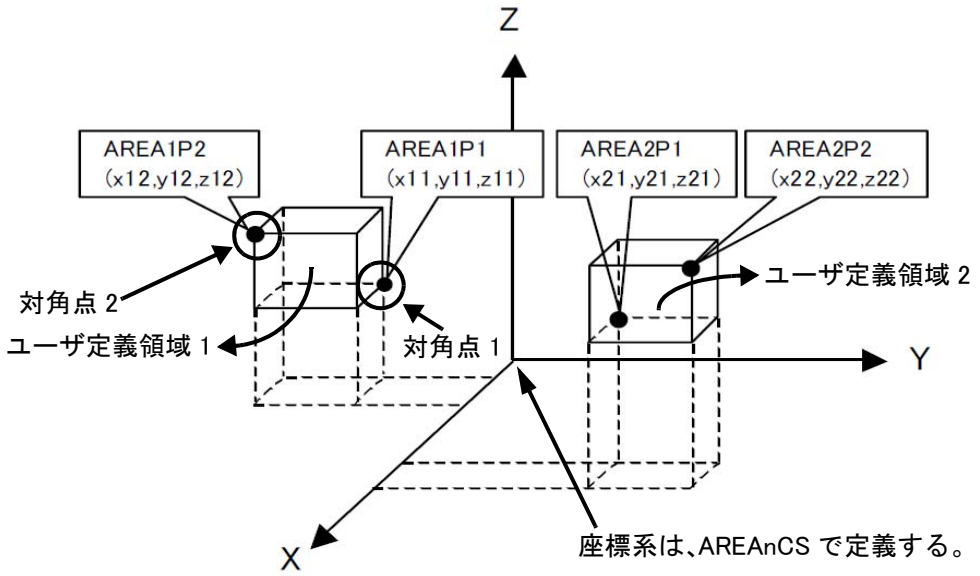
領域には、位置領域と姿勢領域、および付加軸領域があります。
以下にそれぞれの領域について、設定内容の詳細を示します。

(1) 位置領域の設定

AREA*P1・AREA*P2 (*=1 ～ 32) の X,Y,Z 成分で設定された対角点座標で、ユーザ定義領域の位置領域を設定します。設定する座標値は AREA*CS (*=1 ～ 32) で選択された座標系を基準とした値です。

＜注意＞

1) Base 命令等を用いてワールド座標系を変更して使用し、かつユーザ定義領域の座標系の選択で「ベース座標系」を選択される場合、以下の内容に注意して領域を設定してください。
AREAnP1、AREAnP2 の X,Y,Z 成分に設定する座標値は、AREAnCS で選択した座標系での値を設定する必要があります。
T/B や RT ToolBox 等に表示される直交座標値はワールド座標系基準となっています。このため、AREAnCS に「ベース座標系」を選択した場合、表示される座標値と設定すべき座標値は異なるため注意が必要です。
この場合、表示されている座標値をベース座標系に変換して設定するか、あるいは一次的にワールド座標系を初期状態に戻して設定する必要があります。(出荷時は、ベース座標系とワールド座標系は一致しています)
2) ユーザ定義領域の内側または外側の判定は、0.001mm および 0.001 度の単位でおこないます。このため境界線上では判定結果が不定となる場合があります。
3) AREAnP1 と AREAnP2 の X,Y,Z 成分を入れ替えて設定しても定義領域は同じとなります。



(2) 姿勢領域の設定

AREAnP1・AREAnP2 の A,B,C 成分の設定により、ユーザ定義領域の姿勢領域を設定します。
設定する値は AREAnCS で選択された座標系を基準とした値です。

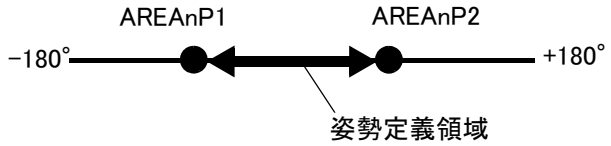
⚠注意

6 軸仕様のロボットをご使用の場合、B 軸の現在座標値が ±90 度付近においては、手首の姿勢移動量が僅かであっても、A 軸および C 軸の座標値は符号が反転されるなど大きく変化します。これはロボットの制御に起因するものです。
このため，ロボットが上記 B 軸 =±90 度付近にある時、A 軸および C 軸の姿勢領域判定結果はロボットの移動量に関係なく変わってしまう場合があるため、本領域判定機能のご利用には適していません。
B 軸の現在座標値が ±90 度付近にならないようなロボット動作において本領域判定機能をご利用いただきますようお願いします

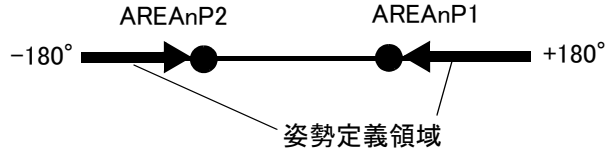
＜注意＞

1) Base 命令等を用いてワールド座標系を変更して使用し、かつユーザ定義領域の座標系の選択で「ベース座標系」を選択される場合、以下の内容に注意して領域を設定してください。
AREAnP1、AREAnP2 の A,B,C 成分に設定する値は、AREAnCS で選択した座標系での値を設定する必要があります。
T/B や RT ToolBox 等に表示される直交座標値はワールド座標系基準となっています。このため、AREAnCS に「ベース座標系」を選択した場合、表示される座標値と設定すべき座標値は異なるため注意が必要です。
この場合、表示されている座標値をベース座標系に変換して設定するか、あるいは一次的にワールド座標系を初期状態に戻して設定する必要があります。(出荷時は、ベース座標系とワールド座標系は一致しています)
2) AREAnP1 と AREAnP2 の A,B,C 各成分の大小関係により、設定する領域が異なります。(下図参照)
3) 姿勢領域をチェックしない場合は、AREAnP1 の A,B,C 成分をすべて -360° に、AREAnP2 の A,B,C 成分をすべて +360° に設定します。
4) ユーザ定義領域の内側または外側の判定は、0.001mm および 0.001 度の単位でおこないます。このため境界線上では判定結果が不定となる場合があります。

【姿勢成分の大小関係が、AREAnP2 > AREAnP1 と設定した場合】



【姿勢成分の大小関係が、AREAnP1 > AREAnP2 と設定した場合】



(3) 付加軸領域の設定

AREAnP1・AREAnP2 の L1,L2 成分の設定により、ユーザ定義領域の付加軸領域を設定します。
付加軸領域を設定した場合は、位置領域・姿勢領域・付加軸領域のすべてが領域内にある場合に、ユーザ定義領域内であると判定します。

＜注意＞

1) AREAnP1、AREAnP2 の L1、L2 成分については、AREAnCS で定義した座標系の影響を受けません。
2) L1、L2 について AREAnP1 と AREAnP2 を入れ替えて設定しても定義領域は同じ領域となります。
3) 付加軸領域を設定した場合は、位置領域・姿勢領域・付加軸領域のすべてが領域内にある場合に、ユーザ定義領域内であると判定します。
4) ユーザ定義領域の内側または外側の判定は、0.001mm および 0.001 度の単位でおこないます。このため境界線上では判定結果が不定となる場合があります。
5) 付加軸 (J7,J8 軸) 使用していない場合、設定は不要です。

5.8.3 対象メカの選択

領域チェックをおこなう対象メカを、パラメータ AREA*ME で指定します。通常は、メカ 1(″1″) を設定します。マルチメカ等を用いている場合は、対応するメカ番号を設定します。

5.8.4 ユーザ定義領域内での挙動指定

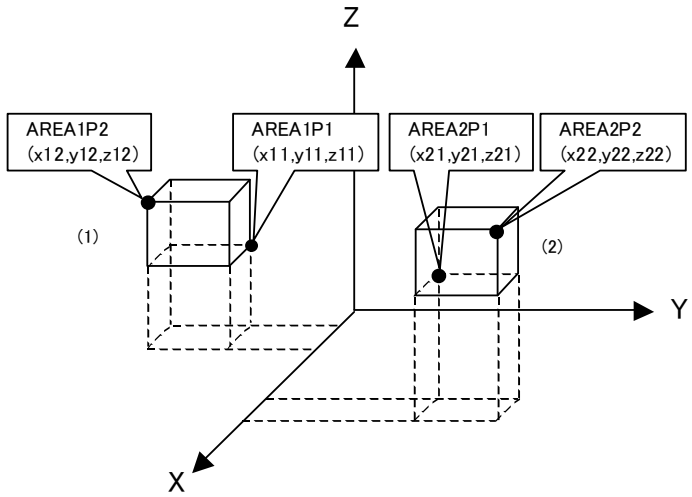
ロボットがユーザ定義領域内にあるかどうかの状態設定を表 5-11 の中から選択し、パラメータ AREAnAT の設定で指定します。

表 5-11：ユーザ定義領域内での挙動指定

設定	領域内	領域外
0：無効	状態設定はおこないません。	状態設定はおこないません。
1：信号出力、及び状態変数設定	専用出力信号 USRAREA が ON します。 システム状態変数 (M_Uar32,M_Uar) の対応 Bit を ON します。	専用出力信号 USRAREA が OFF します。 システム状態変数 (M_Uar32,M_Uar) の対応 Bit を OFF します。
2：エラー出力	エラー：H2090 を発生し停止します。 ※ この場合、位置領域のチェックのみをおこない姿勢領域、付加軸領域は無視します。 ※ ロボットを領域外へ移動させるには、解除できないエラーの一時的エラーリセット操作でジョグ送りにより移動させてください。	-

5.8.5 設定例

下図の領域 (1) で作業中に出力信号番号 10 を、領域 (2) で作業中に出力信号番号 11 番を出力させたい場合は以下のようなパラメータ設定になります。



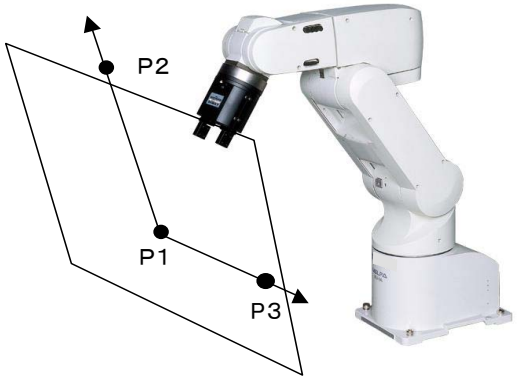
＜領域 1＞
座標系：ワールド座標系
姿勢チェック不要
メカ 1 使用
付加軸未使用
＜領域 2＞
座標系：ワールド座標系
姿勢チェック不要
メカ 1 使用
付加軸未使用

パラメータ名	値の意味	値
AREA1CS	領域 1 の座標系選択	0
AREA1P1	領域 1 の対角点 1 位置データ：X,Y,Z,A,B,C,L1,L2	x11,y11,z11,−360,−360,−360,0,0
AREA1P2	領域 1 の対角点 2 位置データ：X,Y,Z,A,B,C,L1,L2	x12,y12,z12,+360,+360,+360,0,0
AREA1ME	領域 1 の対象メカ番号：通常は 1	1
AREA1AT	領域 1(無効 / 信号出力 / エラー)：0/1/2	1
AREA2CS	領域 2 の座標系選択	0
AREA2P1	領域 2 の対角点 1 位置データ：X,Y,Z,A,B,C,L1,L2	x11,y11,z11,−360,−360,−360,0,0
AREA2P2	領域 2 の対角点 2 位置データ：X,Y,Z,A,B,C,L1,L2	x12,y12,z12,+360,+360,+360,0,0
AREA2ME	領域 2 の対象メカ番号：通常は 1	1
AREA2AT	領域 2(無効 / 信号出力 / エラー)：0/1/2	1
USRAREA	出力信号：開始番号、終了番号	10,11

5.9 自由平面リミット

ワールド座標系内に任意の平面を定義してその平面手前か奥かの判定をおこない、自由平面リミットエラーを出せます。

5.9.1 自由平面リミットの定義



注）図は垂直 6 軸ロボットの例です。水平多関節型ロボットでも同じです。

左図のように 3 点 (P1, P2, P3) にて任意の平面を定義しその面の手前 (ロボットの原点ある側) か、そうでないかを判定します。
本機能によって床面衝突や周辺装置への干渉を防げます。
最大 8 面まで監視できます。
平面の限界はありません。

パラメータと値	説明
SFCnP(n=1 ～ 8)	平面を作成する為の 3 点を指定します。 P1 の座標値の　X1,Y1,Z1：平面の原点位置 P2 の座標値の　X2,Y2,Z2：平面の X 軸上の位置 P3 の座標値の　X3,Y3,Z3：平面の X-Y 平面上の +Y 方向の位置
SFCnME(n=1 ～ 3)	自由平面リミットを有効にするメカ番号を指定します。　通常は 1 を設定します。 マルチメカの場合はそのメカ番号を指定します。
SFCnAT(n=1 ～ 8)	設定した自由平面リミットの有効 / 無効を指定します。 0：無効 (初期値) 1：有効 (動作可能領域はロボット座標原点側) -1：有効 (動作可能領域はロボット座標原点が無い側)

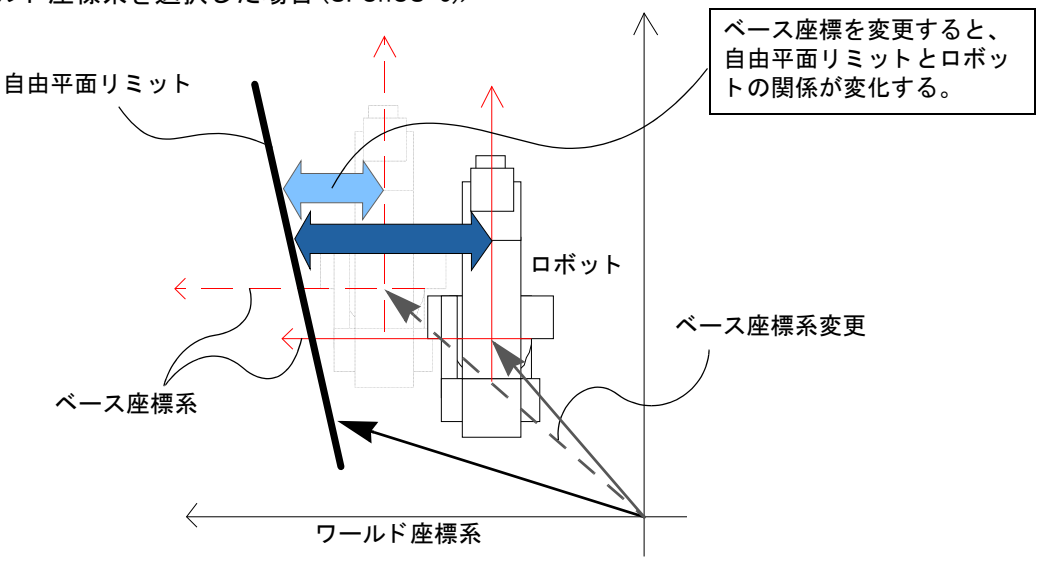
上記パラメータを設定したらコントローラの電源を再投入してください。
以上で平面を越えた場合は自由平面リミットエラーがでるようになります。

5.9.2 自由平面リミットの座標系選択

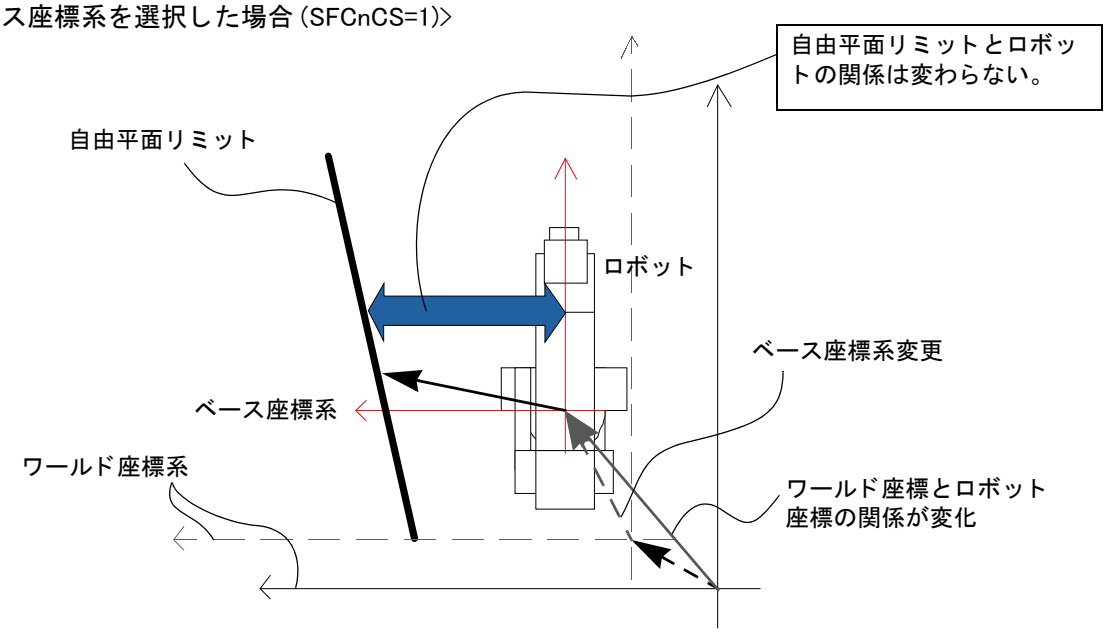
この機能は、Base 命令等を用いてベース座標系を変更してご使用になる場合、自由平面リミットも同時に移動させるか、固定しておくかを選択するものです。選択はパラメータ：SFCnCS にて、基準とする座標系を「ワールド座標系」（移動させる）、または「ベース座標系」（固定しておく）のいずれかを指定します。
ベース座標系を変更しない場合は、どちらを選択しても自由平面リミットは変化しません。
座標系の詳細については、147 ページの「4.5 ロボットの座標系解説」を参照願います。

座標系	説明
ワールド座標系 SFCnCS=0	ベース座標系を変更すると、自由平面リミットも同時に移動します。ロボット本体と自由平面リミットの相対位置関係が変わります。
ベース座標系 SFCnCS=1	ベース座標系を変更しても自由平面リミットは移動しません。ロボット本体と自由平面リミットの相対位置関係は固定です。 ベース座標系を変更して使用するが、ロボット本体と自由平面リミットの相対位置関係を固定させておきたい場合に有効です。

＜ワールド座標系を選択した場合（SFCnCS=0）＞



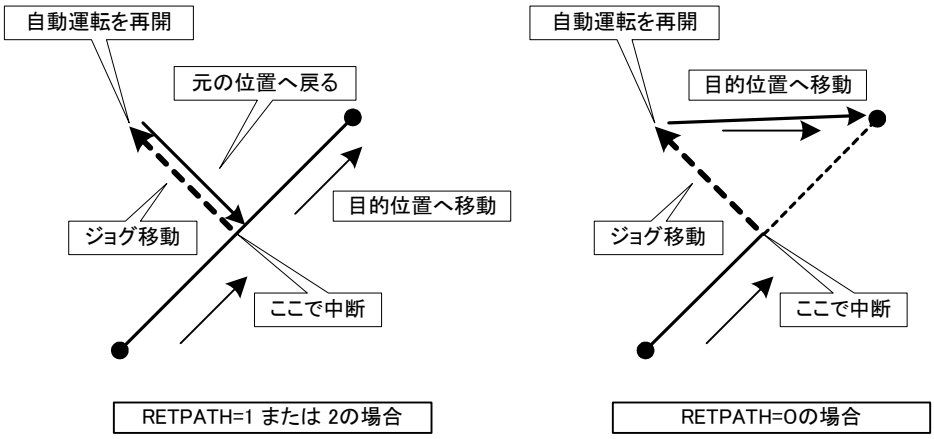
＜ベース座標系を選択した場合（SFCnCS=1）＞



5.10 中断時のジョグ送り後の自動復帰設定

ロボットを自動運転中またはステップ送り中に中断して T/B にて別の位置へジョグ送りしてその位置から自動運転を再開したりステップ送りを再度実行した場合の軌跡の挙動を指定します。下記図を参考にしてください。

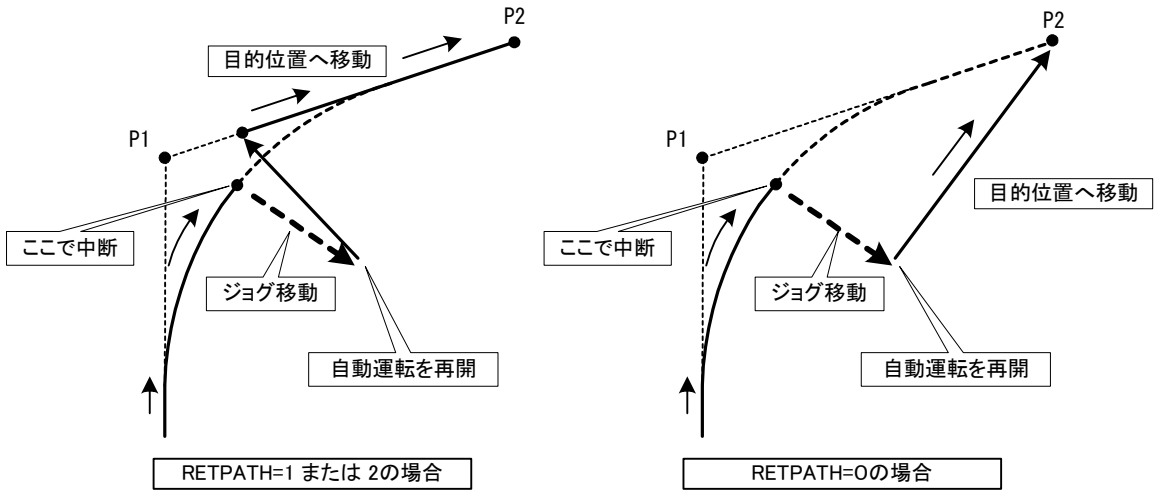
パラメータと値	動作説明
RETPATH=1(デフォルト)	1) 元の中断位置まで関節補間で戻ります。 2) 中断した行を再開します。
RETPATH=0	ジョグで移動した位置から中断行を再開します、よって現在の位置から次の目的位置まで実行中の命令の補間形式で移動します。
RETPATH=2	1) 元の中断位置まで直線補間で戻ります。 2) 中断した行を再開します。



[注意] パラメータ「RETPATH」が 1 の設定にて関節ジョグ以外 (直交、ツール、円筒など) にて移動させた場合は中断位置へ戻るとき関節補間にて動作しますので周辺装置への干渉に十分ご注意願います。

[注意] パラメータ「RETPATH」が 2 の設定のとき、構造データが有効なロボットや多回転を持つロボットの場合、中断位置から関節ジョグで移動させると、本来の構造データや多回転データと異なる位置となり、中断位置へ戻ることができない場合があります。その場合は、中断時の位置に合わせてから再開させてください。

Cnt 命令を使用して連続動作 (軌跡接続動作) を行っている場合は、下図に示すように「RETPATH=1 または 2」と設定されている時は、中断位置ではなく P1 から P2 への移動経路上の位置へ戻ります。
「RETPATH=0」と設定されている時は、現在の位置から目的位置へ移動します。



5.11 電源ONでのプログラムの自動実行方法

コントローラの電源を入れただけでロボットプログラムを自動的に実行する方法を以下に示します。
ただし、電源 ON のみでロボットが動作しますので運用の際には十分ご注意ください。

表 5-12：関連するパラメータ

パラメータと値	動作説明
SLT*	例)SLT2=2,ALWAYS,REP プログラム名、起動条件、運転形態を指定します。 今回は起動条件がポイントです。
ALWENA	0 → 1 常時実行プログラム内で XRun、Xload などのマルチタスク関連の命令や、Servo 命令を実行可能にします。

1) まず、常時実行用のプログラムと動作用プログラムを作成します。
＜プログラム 2 番　常時実行用プログラム＞

```
1 ' Auto Start Sample Program
2 '
3 '  コントローラのモードが"AUTOMATIC"(操作パネル)の時、Prg.1 を実行します。
4 '  "AUTOMATIC"(操作パネル)以外の時、プログラムを停止し実行行を先頭にもどします。
5 '
6 IF M_Mode<>2 And (M_Run(1)=1 Or  M_Wai(1)=1) Then GoSub *MTSTOP
7 IF M_Mode=2  And  M_Run(1)=0 And M_Wai(1)=0  Then GoSub *MTSTART
8 IF M_Mode=1 Then Hlt ' for DEBUG
9 End
10 '
11 *MTSTART
12 XRun 1,"1"
13 Return
14 '
15 *MTSTOP
16 XStp 1
17 Wait M_Run(1)=0
18 XRst 1
19 Return
```

＜プログラム 1 番　動作用プログラム＞(内容は任意です)

```
1 'Main Program(位置データは参考値です)
2 Servo On
3 M_Out(8)=0
4 Mov P1
5 M_Out(8)=1
6 Mov P2
7 End
P1=(+300.00,-200.00,+200.00,+0.00,+180.00,+0.00)(6,0)
P2=(+300.00,+200.00,+200.00,+0.00,+180.00,+0.00)(6,0)
```

2) パラメータを設定します。

パラメータ	動作説明
SLT2	SLT2=2,ALWAYS,REP　　' プログラム 2 番を常時実行モードで実行します。
ALWENA	0 → 1 常時実行プログラム内で XRun、Xload などのマルチタスク関連の命令や、Servo 命令を実行可能にします。

設定後、コントローラの電源を一度 OFF にしてください。

3) 電源を入れます。
上記サンプル例ではコントローラの電源を入れ、コントローラのモードを "AUTOMATIC" にするとプログラム 1 番が実行されロボットが動作します。

5.12 ハンドタイプについて

(1) 電磁弁のタイプと信号番号

ご使用になる電磁弁のタイプと、接続する出力信号に合わせてパラメータを設定します。
設定できる内容を以下に示します。

a) 電磁弁のシンクタイプ / ソースタイプの設定 ..... パラメータ：HIOTYPE
注）本パラメータの設定は、ハンド入力信号のロジックも同時にシンクタイプ、またはソースタイプに設定されます。

b) 電磁弁のシングルソレノイドタイプ / ダブルソレノイドタイプの設定 ..... パラメータ：HANDTYPE

c) 電磁弁を駆動する出力信号番号 ..... パラメータ：HANDTYPE

工場出荷時は、シンクタイプ、かつダブルソレノイドタイプの電磁弁をご使用されることを想定した設定にしています。
この設定とは異なる電磁弁をご使用される場合や、汎用出力信号をご利用になる場合は、表 5-13 に示すパラメータを変更してください。

表 5-13：工場出荷時のパラメータ設定

パラメータ名	値・説明
HIOTYPE	工場出荷設定値：1 電磁弁のシンクタイプ / ソースタイプの設定と、ハンド入力信号ロジックのシンクタイプ / ソースタイプを設定します。 1：シンクタイプ 0：ソースタイプ
HANDTYPE	工場出荷設定値：D900,D902,D904,D906, , , , 電磁弁のシングルソレノイドタイプ / ダブルソレノイドタイプと、電磁弁を駆動する出力信号番号を設定します。 値の左側から順に、ハンド番号 1、2.. に対応しています。初期値は以下のとおりです。 ハンド 1 = 信号番号 900、901 をアクセス ハンド 2 = 信号番号 902、903 をアクセス ハンド 3 = 信号番号 904、905 をアクセス ハンド 4 = 信号番号 906、907 をアクセス このハンド 1 ～ 4(または 8) は命令 (HOpen、HClose) の引数として使用します。 < 設定方法 > ダブルソレノイドタイプをご使用の場合は信号番号の前に 'D' を付けて番号を指定します。 ダブルソレノイドの場合はハンド 1 ～ 4 になります。 シングルソレノイドタイプをご使用の場合は信号番号の前に 'S' を付けて番号を指定します。 シングルソレノイドの場合はハンド 1 ～ 8 になります。 < 設定例 > ※ 下記は、いずれもお客様ご用意のバルブを外部配線にて汎用出力信号に接続した例です。 ①ダブルソレノイドタイプで汎用信号 10 番からハンド 2 つ分を割り付ける場合 HANDTYPE=D10,D12, , , , , ②シングルソレノイドタイプで汎用信号 10 番からハンド 3 つ分を割り付ける場合 HANDTYPE=S10,S11,S12, , , , , ③ハンド 1 番にダブルソレノイドで 10 番を割り付け、ハンド 2 番にシングルソレノイドで 12 番を割り付ける場合 HANDTYPE=D10,S12, , , , ,

5.13 ハンド初期状態について

(1) CR750/CR751 コントローラ

工場出荷時の設定は、以下のようになっています。

ハンドの種類	状態	出力信号番号の状態
エアーハンドのダブルソレノイドでの駆動を想定	ハンド 1 = 開	900=1 901=0
	ハンド 2 = 開	902=1 903=0
	ハンド 3 = 開	904=1 905=0
	ハンド 4 = 開	906=1 907=0

電源投入時、全てのハンドは〝開〟の状態で立ち上がるように初期パラメータを以下のように設定しています。

パラメータ名	値 900, 901, 902, 903, 904, 905, 906, 907
HANDINIT	1, 0, 1, 0, 1, 0, 1, 0

標準構成(1 台接続)の場合は上記のとおりです。

例えば、電源 ON 時に必ずハンド 1 のみを閉じたい場合は次のように設定します。

パラメータ名	値
HANDINIT	0, 1, 1, 0, 1, 0, 1, 0, 1

[注意]ハンド初期状態を「開」に設定しておくと電源 ON 時にワークを落とす場合がありますので十分にご注意ください。

[注意]本パラメータはロボットの先端にあるハンド専用信号(900 番台)の電源 ON 時の初期値を指定するものです。
ハンド制御を汎用 IO(900 番台以外)で制御する場合(HANDTYPE パラメータにて 900 番以外の信号を指定)の電源 ON 時の初期状態を設定するには、この HANDINIT パラメータではなく、ORS* パラメータを使用してください。この ORS* パラメータで設定された値が電源 ON 時の信号の初期値になります。

(2) CR760 コントローラ

工場出荷時の設定は、以下のようになっています。

ハンドの種類	状態	出力信号番号の状態
エアーハンドのダブルソレノイドでの駆動を想定	ハンド 1 = 開	764=0 765=0
	ハンド 2 = 開	766=0 767=0
	ハンド 3 = 開	768=0 769=0
	ハンド 4 = 開	770=0 771=0

種類	パラメータ名	値
ハンド出力信号	ORST764	信号番号 764----771 772----779 780----787 788----795 | | | | | | | | ********, ********, ********, ********

各値は、左から順にビット対応しています。設定する値は『0』または『1』または『*』の 3 種類です。

『0』= OFF に設定

『1』= ON に設定

『*』= は現状維持、変更されません。(電源 ON 時は OFF となります)

例えば、電源 ON 時に必ずハンド 1 のみを閉じたい場合は次のように設定します。

パラメータ名	値
ORST764	信号番号 764----771 772----779 780----787 788----795 | | | | | | | | 01******, ********, ********, ********

[注意] ハンド初期状態を「開」に設定しておくと電源 ON 時にワークを落とす場合がありますので十分にご注意ください。

5.14 出力信号リセットパターンについて

工場出荷時の設定は、全ての汎用出力信号が OFF(0) で立ち上がるようになっています。電源 ON 時の汎用出力信号の状態を変更するためには以下のパラメータを変更することにより可能です。このパラメータは、汎用出力信号リセット操作 (専用入力信号などで実行するもの)、Clr 命令実行時のリセットパターンと兼用しますので注意してください。

種類	パラメータ名	値 (工場出荷時は全て 0 になっています。)
リモートIO/パラレルIO	ORST0	信号番号 0------7 8-----15 16----23 24----31 00000000、00000000、00000000、00000000
	ORST32	32----40 41----49 50----57 58----66(以下同様) 00000000、00000000、00000000、00000000
	ORST96	00000000、00000000、00000000、00000000
	ORST64	00000000、00000000、00000000、00000000
	ORST128	00000000、00000000、00000000、00000000
	ORST160	00000000、00000000、00000000、00000000
	ORST192	00000000、00000000、00000000、00000000
	ORST224	00000000、00000000、00000000、00000000
PROFIBUSオプション	ORST2000	00000000、00000000、00000000、00000000
	ORST2032	00000000、00000000、00000000、00000000
	ORST2064	00000000、00000000、00000000、00000000
	ORST2096	00000000、00000000、00000000、00000000
	ORST2128	00000000、00000000、00000000、00000000
	ORST2160	00000000、00000000、00000000、00000000
	ORST2192	00000000、00000000、00000000、00000000
	ORST2224	00000000、00000000、00000000、00000000
	ORST2256	00000000、00000000、00000000、00000000
	ORST2288	00000000、00000000、00000000、00000000
	:	:
	:	:
	:	:
	ORST5008	00000000、00000000、00000000、00000000
	ORST5040	00000000、00000000、00000000、00000000
CC-LINKオプション	ORST6000	00000000、00000000、00000000、00000000
	ORST6032	00000000、00000000、00000000、00000000
	ORST6064	00000000、00000000、00000000、00000000
	ORST6096	00000000、00000000、00000000、00000000
	ORST6128	00000000、00000000、00000000、00000000
	ORST6160	00000000、00000000、00000000、00000000
	ORST6192	00000000、00000000、00000000、00000000
	ORST6224	00000000、00000000、00000000、00000000
	ORST6256	00000000、00000000、00000000、00000000
	ORST6288	00000000、00000000、00000000、00000000
	:	:
	:	:
	:	:
	ORST7984	00000000、00000000、00000000、00000000
	ORST8016	00000000、00000000、00000000、00000000

種類	パラメータ名	値（工場出荷時は全て 0 になっています。）
シーケンサリンク 注1)	ORS10000	信号番号 10000---10007 10008---10015 10016---10023 10024---10031 | | | | | | | | 00000000、 00000000、 00000000、 00000000
	ORS10032	10032---10039 10040---10047 10048---10055 10056---10063 | | | | | | | | 00000000、 00000000、 00000000、 00000000
	ORS10064	00000000、00000000、00000000、00000000
	ORS10096	00000000、00000000、00000000、00000000
	ORS10128	00000000、00000000、00000000、00000000
	|	00000000、00000000、00000000、00000000
	|	00000000、00000000、00000000、00000000
	|	00000000、00000000、00000000、00000000
	|	00000000、00000000、00000000、00000000
	|	00000000、00000000、00000000、00000000
	ORS18160	00000000、00000000、00000000、00000000

注 1）シーケンサリンクは CR7xx-Q シリーズのみです。

各値は、左から順にビット対応しています。設定する値は『0』また『1』または『*』の 3 種類です。
『0』= OFF に設定
『1』= ON に設定
『*』= は現状維持、変更されません。（電源 ON 時は OFF となります）
例えば、電源 ON 時には必ず汎用信号の 10138、10139、10140 番および汎用出力信号 10160、10161、10168 を ON の状態にしたい場合は次のように設定します。

パラメータ名	値
ORS10128	10128---10135 10136---10143 10144---10151 10152---10159 | | | | | | | | 00000000、 00000000、 00000000、 00000000..... 出荷時の値 00000000、 00**111**000、 00000000、 00000000..... 今回の設定値
ORS10160	10160---10167 10168---10175 10176---10183 10184---10191 | | | | | | | | 00000000、 00000000、 00000000、 00000000..... 出荷時の値 **11**000000、 **1**0000000、 00000000、 00000000..... 今回の設定値

上記に加えて、汎用出力信号リセット操作時には 10148、10149、10150 は ON なら ON のまま、OFF なら OFF のままという現状維持でリセットしないようにする場合には、次のように設定します。

パラメータ名	値
ORS10128	00000000、00111000、0000***0、00000000
ORS10160	11000000、10000000、00000000、00000000

上記の場合は、電源 ON 時には 10148、10149、10150 は 0(OFF) で立ち上がります。電源 ON 時には 1(ON) にして、汎用出力信号リセット操作時には現状維持という設定はできません。

[注意] パラメータを編集する場合はゼロの個数を間違えないようにしてください。ゼロの個数に過不足があると次回の電源 ON 時にエラーが発生します。

5.15 通信設定について(イーサネット)

通信に使用するポート番号を表 5-14 に示します。

表 5-14 : 使用ポート番号

ポート番号	用途	パラメータ注 1)
20	予約	
21	予約	
23	COGNEX ビジョンセンサ	変更不可
111	予約	
502	予約	
1024	予約	
5001	GOT 通信	GOTPORT
10000	Mxt 命令	NETPORT
10001	RT ToolBox2(データリンクでも使用可能)	NETPORT
10002 ～ 10009	データリンク (RT ToolBox2 でも使用可能)	NETPORT
11000	R56TB/R57TB	変更不可 (TBPORT)
12000	オシログラフ機能	MONPORT
43815	予約	
43816	予約	

注 1) ポート番号は、一部を除いてパラメータで変更することができます。設定範囲は 0 ～ 65535 です。

5.15.1 各パラメータの詳細

各パラメータの詳細を以下に示します。

(1) NETIP(コントローラの IP アドレス)

コントローラの IP アドレスを設定します。IP アドレスとは、郵便物の住所のようなものです。IP アドレスの形式は、0 ～ 255 までの 4 つの数字とその間のドット (.) で表記します。たとえば、192.168.0.1 や、10.97.11.31 というように設定します。
コントローラとネットワークパソコンとを 1 対 1 で直結する場合は、IP アドレスは初期値 (任意のもの) で構いませんが、ローカルエリアネットワーク (LAN) に接続する場合には、お客様の LAN システムの管理者にしたがって IP アドレスを設定してください。IP アドレスが重複した場合は、機能がただしく動作しませんので、設定する際は、他のものと重複しないようにご注意ください。
なおコントローラと通信するパソコンは、同一ネットワーク上にあるとことが必要です。

(2) NETMSK(サブネットマスク)

コントローラのサブネットマスクを設定します。サブネットマスクとは、IP アドレスのうち、サブネットワークを定義するために設定するものです。サブネットマスクの形式は、0 ～ 255 までの 4 つの数字とその間のドット (.) で表記します。
たとえば、255.255.255.0 や、255.255.0.0 というように設定します。通常は、初期値のままで構いません。ローカルエリアネットワーク (LAN) に接続する場合には、お客様の LAN システムの管理者にしたがってサブネットマスクを設定してください。

(3) NETPORT(ポート番号)

コントローラのポート番号を設定します。ポート番号は郵便物の氏名のようなものです。9 つの要素について、ポート番号をそれぞれ数値で表します。1 つめ (要素番号 1) は、リアルタイム制御用、2 つめから 9 つめ (要素番号 2 ～ 9) までは、サポートソフト用またはデータリンク用です。通常は初期値のままで変更する必要はありません。また、ポート番号が重複しないようにしてください。

(4) CPRCE11 ～ 19(プロトコル)

データリンク機能を使用する場合、必ず設定が必要です。通信する際のプロトコル (手順) を設定します。プロトコルは、無手順、有手順、データリンクの 3 種類あります。

設定値は、以下の 0 ～ 2 に対応します。

0... 無手順 : パソコンサポートソフトを使用する際のプロトコルです。

1... 有手順 : 予約 （機能がありませんので、誤って設定しないでください）

2... データリンク : ロボットプログラムの Open/Input/Print 命令を使って通信する際のプロトコルです。

(5) COMDEV(COM1 : ～ 8 に対応するデバイス定義)

データリンク機能を使用する場合、必ず設定が必要です。COM1 : ～ 8 に対応するデバイス定義を設定します。COM1 : ～ 8 は、ロボットプログラムの Open 命令で使います。プロトコルの設定 (CPRCE11 ～ 19) で、データリンクを指定した場合のみ、必ず設定してください。イーサネットインタフェースオプションでは、OPT11 ～ OPT19 のデバイス名 (文字列) を設定します。OPT11 ～ OPT19 の設定値は、下表のようにパラメータ NETPORT で設定したポート番号に対応しています。

※ 以下のパラメータ NETOPORT(n) および COMDEV(n) で、数値 n は同パラメータの要素番号を表します。

数値 n	″COMDEV(n) で設定するデバイス名″	ポート番号
1	OPT11	NETPORT(2) で指定したポート番号
2	OPT12	NETPORT(3) で指定したポート番号
3	OPT13	NETPORT(4) で指定したポート番号
4	OPT14	NETPORT(5) で指定したポート番号
5	OPT15	NETPORT(6) で指定したポート番号
6	OPT16	NETPORT(7) で指定したポート番号
7	OPT17	NETPORT(8) で指定したポート番号
8	OPT18	NETPORT(9) で指定したポート番号
9	OPT19	NETPORT(10) で指定したポート番号

たとえば、NETPORT(3) で指定したポート番号を、通信回線ファイル名 COM3 : としてデータリンクに割り当てる場合は、

COMDEV(3) = OPT13 ※ COMDEV の 3 要素めに OPT13 を設定する

CPRCE13 = 2 ※ データリンクとして設定する

とします。

(6) NETMODE(サーバー指定)

データリンク機能を使用する場合に設定します。

コントローラのデータリンク機能における TCP/IP 通信をサーバーまたはクライアントに設定します。コントローラに接続する相手側のアプリケーションによって切り替える必要があります。

(7) NETHSTIP(データ通信先のサーバーの IP アドレス)

データリンク機能でコントローラをクライアントにして使用する場合に設定します。データリンク機能で、コントローラが接続する相手サーバーの IP アドレスを指定します。NETMODE のサーバー指定で、コントローラを「クライアント」にしたときのみ設定します。

(8) MXTTOUT(リアルタイム外部制御命令実行時のタイムアウト設定)

リアルタイム外部制御命令を使い、コントローラとの通信タイムアウト時間を設定したいときに変更します。1 制御周期の約 7.11msec の倍数を指定します。リアルタイム外部制御命令が実行されているとき、コントローラに対してパソコンから通信データが来ないときタイムアウト時間をカウントアップし、この MXTTOUT で設定したカウントになるとエラー (H7820) で停止するようになります。例えば、約 7 秒の間に通信が無い場合エラーにしたいときは、1000 に設定します。工場出荷時には、タイムアウトなし (-1) に設定されています。

5.15.2 パラメータ設定例1(ｻﾎﾟｰﾄｿﾌﾄｳｪｱを使う場合)

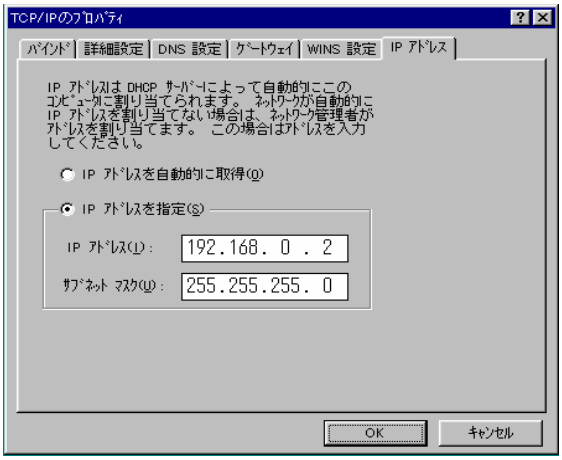
サポートソフトを使う場合の設定例を以下に示します。
コントローラのパラメータ設定と、ご使用になるパソコンの OS のネットワーク設定をおこないます。

項目	設定値
コントローラの IP アドレス	192.168.0.20
パソコンの IP アドレス	192.168.0.2
コントローラのポート番号	10001

下表のようにコントローラのパラメータを設定します。
工場出荷時の設定のままであればパラメータを変更することなくそのままご使用できます。

変更するパラメータ名	変更前・後	パラメータの値
NETIP	前	192.168.0.20
	後	〃　（初期値のまま）
NETPORT	前	10001
	後	〃　（初期値のまま）

つぎにパソコンの IP アドレスを 192.168.0.2 に設定してください。ネットワークのプロパティで設定します。参考までに Windows95(左下図)、WindowsNT4.0(右下図) を示します。



パソコンの IP アドレスは、Windows の TCP/IP プロパティのネットワークの設定 (ネットワークコンピュータのプロパティの中) でおこないます。Windows のバージョンによって設定画面が異なりますので、詳しい設定方法については、Windows 付属のマニュアル等を参照してください。
また、パソコンサポートソフトの設定方法や使い方については、パソコンサポートソフトウェアに付属の取扱説明書を参照してください。

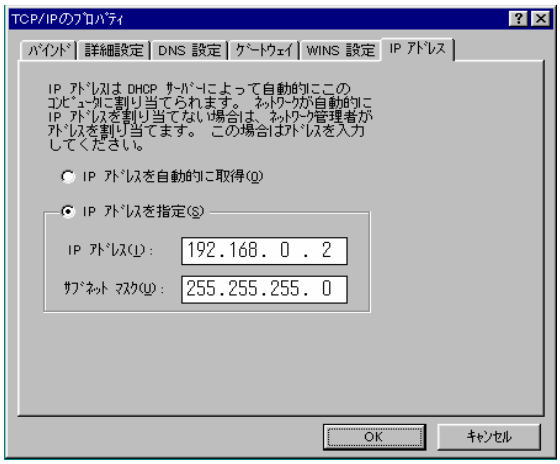
5.15.3 パラメータ設定例2-1(ﾃﾞｰﾀﾘﾝｸ機能を使用：ｺﾝﾄﾛｰﾗがｻｰﾊﾞの場合)

データリンク機能でコントローラがサーバーの場合の設定例を以下に示します。

項目	設定値
コントローラの IP アドレス	192.168.0.20
パソコンの IP アドレス	192.168.0.2
コントローラのポート番号	10003
通信回線番号 OPEN 命令の COM 番号	COM3

変更するパラメータ名	変更前・後	パラメータの値
NETIP	前	192.168.0.20
	後	〃 （初期値のまま）
NETPORT	前	10000,10001,10002,10003,10004,10005,10006,10007,10008,10009
	後	〃 （初期値のまま）
CPRCE13	前	0
	後	**2**
COMDEV	前	RS232, , , , , , ,
	後	RS232, , **OPT13**, , , , ,

パソコンの IP アドレスを 192.168.0.2 に設定してください。ネットワークのプロパティで設定します。参考までに Windows95(左下図)、WindowsNT4.0(右下図) を示します。



パソコンの IP アドレスは、Windows の TCP/IP プロパティのネットワークの設定 (ネットワークコンピュータのプロパティの中) でおこないます。Windows のバージョンによって設定画面が異なりますので、詳しい設定方法については、Windows 付属のマニュアル等を参照してください。

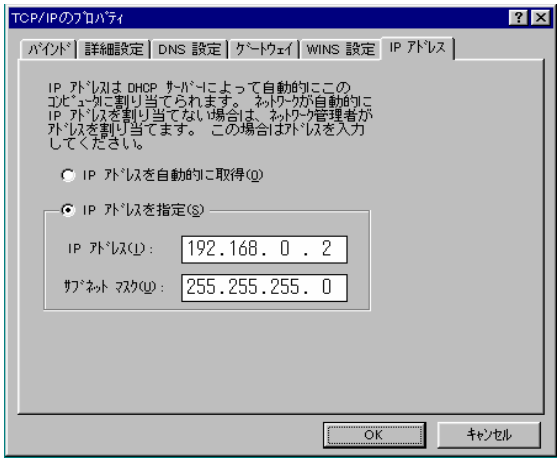
5.15.4 パラメータ設定例2-2(ﾃﾞｰﾀﾘﾝｸ機能を使用：ｺﾝﾄﾛｰﾗがｸﾗｲｱﾝﾄの場合)

データリンク機能でコントローラがクライアントの場合の設定例を以下に示します。

項目	設定値
コントローラの IP アドレス	192.168.0.20
パソコンの IP アドレス	192.168.0.2
コントローラのポート番号	10003
通信回線番号 OPEN 命令の COM 番号	COM3

変更するパラメータ名	変更前・後	パラメータの値
NETIP	前	192.168.0.20
	後	〃 （初期値のまま）
NETPORT	前	10000,10001,10002,10003,10004,10005,10006,10007,10008,10009
	後	〃 （初期値のまま）
CPRCE13	前	0
	後	**2**
COMDEV	前	RS232, , , , , ,
	後	RS232, , **OPT13**, , , ,
NETMODE	前	1,1,1,1,1,1,1,1,1,1
	後	1,1,**0**,1,1,1,1,1,1
NETHSTIP	前	192.168.0.2, 192.168.0.3, 192.168.0.4, 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9, 192.168.0.10
	後	192.168.0.2, 192.168.0.3, **192.168.0.2**, 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9, 192.168.0.10

パソコンの IP アドレスを 192.168.0.2 に設定してください。ネットワークのプロパティで設定します。参考までに Windows95(左下図)、WindowsNT4.0(右下図) を示します。



パソコンの IP アドレスは、Windows の TCP/IP プロパティのネットワークの設定 (ネットワークコンピュータのプロパティの中) でおこないます。Windows のバージョンによって設定画面が異なりますので、詳しい設定方法については、Windows 付属のマニュアル等を参照してください。

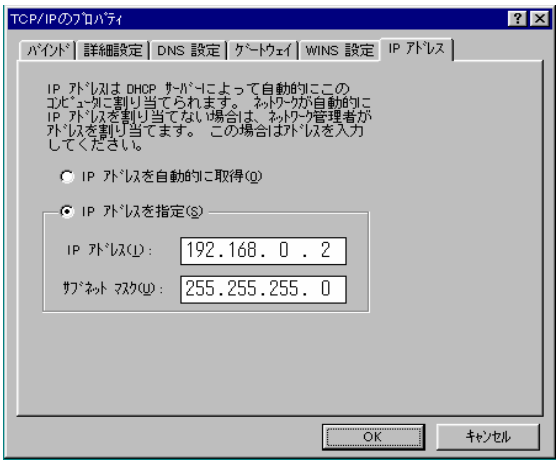
5.15.5 パラメータ設定例3(リアルタイム外部制御機能を使用)

リアルタイム外部制御機能を使う場合の設定例を以下に示します。

項目	設定値
コントローラの IP アドレス	192.168.0.20
パソコンの IP アドレス	192.168.0.2
コントローラのポート番号	10000

変更するパラメータ名	変更前・後	パラメータの値
NETIP	前	192.168.0.20
	後	〃 （初期値のまま）
NETPORT	前	10000,10001,10002,10003,10004,10005,10006,10007,10008,10009
	後	〃 （初期値のまま）
MXTTOUT	前	-1
	後	〃 （初期値のまま）

パソコンの IP アドレスを 192.168.0.2 に設定してください。ネットワークのプロパティで設定します。参考までに Windows95(左下図)、WindowsNT4.0(右下図) を示します。



パソコンの IP アドレスは、Windows の TCP/IP プロパティのネットワークの設定 (ネットワークコンピュータのプロパティの中) でおこないます。Windows のバージョンによって設定画面が異なりますので、詳しい設定方法については、Windows 付属のマニュアル等を参照してください。

5.15.6 接続の確認

ご使用になる前に以下の項目を再度確認してください。

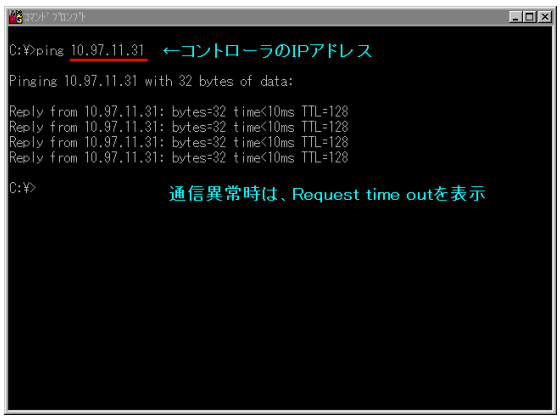
番号	確認項目	チェック
1	ティーチングボックスは確実に固定されていますか？	
2	コントローラとパソコン間のイーサネットケーブルは正しく接続されていますか？	
3	イーサネットケーブルは正しいものを使っていますか？（パソコンとコントローラを 1 対 1 で直結する場合は、クロスケーブルです。LAN でハブを使う場合は、ストレートケーブルにします）	
4	コントローラのパラメータは正しく設定されていますか？	
5	パラメータ設定後、一度コントローラの電源を切りましたか？	

5.15.7 Windowsのpingコマンドによる接続の確認方法

Windows の ping コマンドを使って接続の確認をする方法を示します。

Windows の「スタート」-「プログラム(P)」メニューから、MS-DOS プロンプト（コマンドプロンプト）を起動し、以下のようにコントローラの IP アドレスを指定します。

通信が正常の時は、下図のように Reply from … と表示されます。異常の時は、Request Time out と表示されます。



5.16 ハンド、ワーク条件設定(最適加減速設定)について

最適加減速制御は Loadset 命令、Oadl 命令によってロボット先端の負荷に応じた最適な加減速を自動で行うようになっています。最適な加減速を得るためには下記のパラメータを正しく設定しておくことが条件になります。

また、衝突検知機能でも本パラメータを使用します。

ジョグ操作時に衝突検知機能を使用する場合は HNDDAT0,WRKDAT0 を正しく設定してお使いください。

工場出荷時の設定は、以下のようになっています。

	パラメータ	値
ハンドの条件を設定	HNDDAT0	機種によって異なります。
	HNDDAT1	最大負荷、0.0、0.0、0.0、0.0、0.0、0.0
	HNDDAT2	最大負荷、0.0、0.0、0.0、0.0、0.0、0.0
	HNDDAT3	最大負荷、0.0、0.0、0.0、0.0、0.0、0.0
	HNDDAT4	最大負荷、0.0、0.0、0.0、0.0、0.0、0.0
	HNDDAT5	最大負荷、0.0、0.0、0.0、0.0、0.0、0.0
	HNDDAT6	最大負荷、0.0、0.0、0.0、0.0、0.0、0.0
	HNDDAT7	最大負荷、0.0、0.0、0.0、0.0、0.0、0.0
	HNDDAT8	最大負荷、0.0、0.0、0.0、0.0、0.0、0.0
ワークの条件を設定	WRKDAT0	機種によって異なります。
	WRKDAT1	0.0、0.0、0.0、0.0、0.0、0.0、0.0
	WRKDAT2	0.0、0.0、0.0、0.0、0.0、0.0、0.0
	WRKDAT3	0.0、0.0、0.0、0.0、0.0、0.0、0.0
	WRKDAT4	0.0、0.0、0.0、0.0、0.0、0.0、0.0
	WRKDAT5	0.0、0.0、0.0、0.0、0.0、0.0、0.0
	WRKDAT6	0.0、0.0、0.0、0.0、0.0、0.0、0.0
	WRKDAT7	0.0、0.0、0.0、0.0、0.0、0.0、0.0
	WRKDAT8	0.0、0.0、0.0、0.0、0.0、0.0、0.0

パラメータ値は左から順に(重量、大きさ X、Y、Z、重心 X、Y、Z)を設定し、ハンドの条件、ワークの条件それぞれ 8 つまで設定が可能です。ハンドの大きさは直方体にハンドを入れた場合の長さを入れてください。LoadSet 命令で指定されたハンド条件とワーク条件で最適加減速の計算をおこないます。

パラメータ	値(工場出荷時)
HNDHOLD1	0、1
HNDHOLD2	0、1
HNDHOLD3	0、1
HNDHOLD4	0、1
HNDHOLD5	0、1
HNDHOLD6	0、1
HNDHOLD7	0、1
HNDHOLD8	0、1

パラメータ値は左から順に(OPEN 時, CLOSE 時)の把持・未把持を設定します。

『0』= 未把持に設定

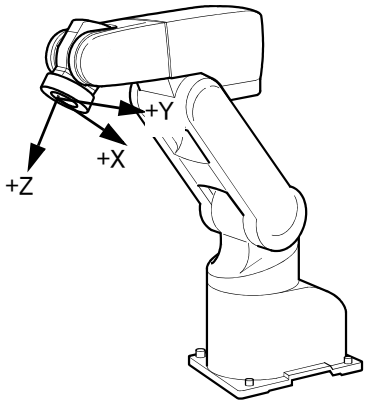
『1』= 把持に設定

・ハンドの開閉状態に応じて、ハンドのみか、ハンドとワークかを切り換えて最適加減速の計算をおこないます。

・ハンドの開閉状態は、HOpen/HClose 命令の実行によって切り換わります。

以下に、ロボットの形式別にハンドの条件、およびワークの条件を設定する際の基準となる座標軸を示します。ハンドの条件、ワークの条件とも座標軸の基準は同じです。なお、大きさは全て正の値で設定します。

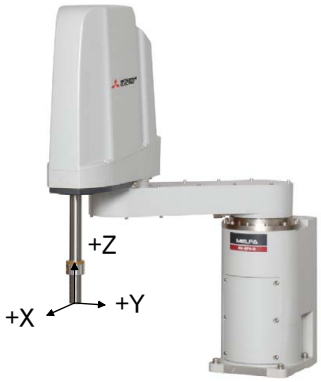
■垂直多関節型ロボット



座標軸の定義
ツール座標を座標軸とします。

必要な設定軸：重心の X, Y, Z 成分と
大きさの X, Y, Z 方向を設定

■水平多関節型ロボット



座標軸の定義
ツール座標を座標軸とします。

必要な設定軸：重心の X, Y, Z 成分と
大きさの X, Y, Z 方向を設定

5.17 特異点近傍警告について

特異点が存在するロボットにおいて、T/B を使ってロボットを動作させている時に、ロボットの制御点が特異点へ近づくと警告を発生し、ロボットを操作している人に注意を促すものです。
警告が発生しても、動作が可能であれば操作を中断しない限りロボットは動作を続けます。また、特異点の近傍から離れたときは自動的に警告をリセットします。以下に、警告についての詳細を示します。

(1) 警告を発生する操作

T/B を使って以下に示す操作をおこなってロボットが動作している時に、ロボットの制御点が特異点へ近づくと警告を発生します。

①ジョグ操作。(関節ジョグモード以外)
②ステップ送り、ステップ戻し操作。
③ MS ポジション移動操作。
④ダイレクト実行操作。

上記のいずれかの操作で特異点へ近づくと、コントローラのブザーが鳴り続けます。(連続音) 但し、操作パネルの STATUS. NUMBER 表示は変化しません。

(2) 警告を発生しない操作

以下に示す操作や動作をおこなっている場合は、ロボットの制御点が特異点に近づいても警告は発生しません。

① T/B を使って、関節ジョグモードから入った付加軸ジョグ操作。
② T/B からの操作でも、関節補間命令を実行している場合。
(Mov 命令などの実行、ポジション移動操作)
③プログラムを自動運転している場合。
④専用入力信号 (JOGENA, JOGM など) によるジョグ操作。
⑤ブレーキを解除して、外力によりロボットを動かしている場合。
⑥ロボットが静止している場合。

5.18 高速RAM運転機能について

高速 RAM 運転機能はプログラムの動作、データの保持に関して一部制限がありますので、仕様を十分ご理解の上ご使用願います。

(1) 概要

ロボットプログラムはバッテリにバックアップされた RAM(SRAM) に保存されます。
そのプログラムの実行には SRAM 運転と DRAM 運転があり、パラメータ：ROMDRV で選択できますが, 出荷時の設定は DRAM 運転になっています。(高速 RAM 運転) DRAM 運転の方が高速に実行できますが (最大で SRAM 運転の 1.2 倍)、プログラムで使用している変数の電源遮断時の保存に一部制約があります。

制約内容

SRAM 運転：全ての変数を保存。
DRAM 運転：プログラム外部変数を保存。ローカル変数とユーザ定義外部変数は破棄。

お客様の用途に合わせて, ロボットプログラムの運転機能 (DRAM 運転, 又は SRAM 運転) を変更してご使用いただけます。
表 5-15 に示すパラメータ：ROMDRV を変更して一度電源を入れ直してください。パラメータの変更は 85 ページの「3.14 パラメータ画面の操作」を参照願います

表 5-15：ROM/ 高速 RAM 運転パラメーター覧

パラメータ	説明と値
ROMDRV	プログラムの実行 (運転) を SRAM 運転 /DRAM 運転に切り替えます。 0= SRAM 運転 2= DRAM 運転 (出荷時設定値)

表 5-16：各メモリの役割と ROMDRV パラメータの関係

メモリ種類	特徴	ROMDRV パラメータ	
		0(SRAM モード)	2(DRAM モード)
SRAM	電源 OFF で消えない 電池消耗で消える 読み書き可能	プログラム実行 (実行結果保存) プログラム管理操作 注 1) システムデータ読み書き 注 2) 共通変数読み書き プログラム読み書き	プログラム管理操作注 1) システムデータ読み書き 注 2) 共通変数読み書き プログラム読み書き
DRAM	高速実行可能 電源 OFF で消える	–	プログラム実行 (実行結果破棄)

注 1）プログラム管理操作とは、T/B やパソコンサポートソフトウェアによりコントローラ内のプログラムのコピー、削除、プログラム名変更などの操作を示します。
注 2）システムデータ保存には「パラメータ保存」、「エラーログ保存」。プログラム (OPEN/PRINT/INPUT) で読み書きするデータファイルが含まれます。

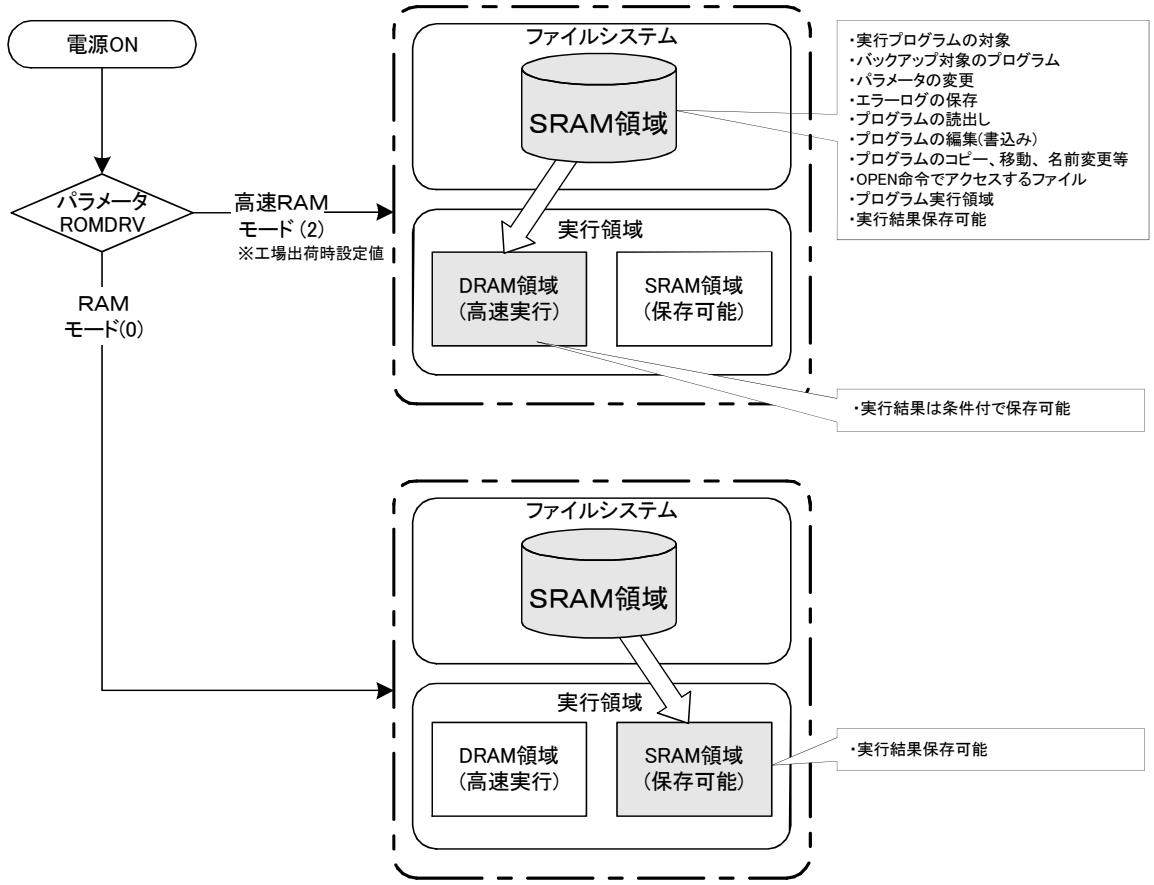


図 5-5：高速 RAM 運転の概念図

高速 RAM 運転中は、DRAM メモリを使用して実行します。SRAM メモリで実行するよりも最大で 1.2 倍程度の高速処理が可能です。(プログラム内容により変化します。)

ただし、プログラムの実行やステップ運転などロボットの操作は SRAM 運転と同様におこなえますが、一部の操作に制限があります。次に示す注意事項を参照願います。

(2) 電源遮断時の変数の保存に関する注意事項

DRAM 運転では、プログラム実行中の変数値の変更は可能ですが、変更された値はコントローラの電源遮断により破棄されますのでご注意願います。以下の方法で電源遮断時に変数の値を保存しておくことができます。

※ 電源断時にも変数値の保存が必要な場合：

プログラム外部変数を使うか、または電源を遮断する前にプログラム選択の実施、または CallP 命令を実行し完了 (CallP 先の End 命令まで実行) させてください。但しユーザ定義外部変数を除きます。

表 5-17：DRAM 運転 /SRAM 運転での変数の保存

変数 注 1)	高速 RAM 運転中 (DRAM 運転)	SRAM 運転中
ローカル変数	電源遮断時に実行していたプログラムの変数値は破棄されます。 プログラム選択時、または CALLP 命令終了時には保存されます。	変数の値は、電源遮断後もそのまま保持。 注 2)
プログラム外部変数	変数の値は、電源遮断後もそのまま保持。	
ユーザ定義外部変数	電源遮断時に実行していたプログラムの変数値は破棄されます。	変数の値は、電源遮断後もそのまま保持。

注 1) それぞれの変数には、数値変数、文字列変数、位置変数、関節変数があります。

注 2) ただし、パソコンサポートソフトウエアでプログラムを書換えた場合、プログラムで使用しているローカル変数の値は破棄されます。

5.19 暖機運転モードについて

(1) 機能の概要

弊社のロボットは、常温の環境下で動作させたとき最も性能を発揮して使用できるように、加減速度やサーボ系などを調整しています。このため、低温下や長期停止後の稼働時には、部品の潤滑に使用しているグリースの粘性の変化により、本来の性能が発揮できず、位置精度の悪化や誤差過大などのサーボエラーが発生する場合があります。このような場合は、低速にてならし運転(暖機運転)をおこなってから本稼働に移行するようお願いしていますが、そのためには暖機運転用のプログラムを別に用意する必要があります。

暖機運転モードとは、コントローラの電源投入直後は速度を下げて運転し、動作時間の経過とともに徐々に元の速度へ戻すことをおこなう機能で、これにより別のプログラムを用意しなくても簡単に暖機運転をおこなうことができます。低温下や長期停止後の稼働時に誤差過大エラーなどが発生する場合は、暖機運転モードを有効としてください。

■暖機運転モードを使用するには

暖機運転モードを使用するには、パラメータ WUPENA に 1(有効)を指定し、コントローラの電源を再投入します。

■暖機運転モードが有効のとき

暖機運転モードが有効のとき、コントローラの電源を投入すると暖機運転状態(速度を自動的に下げている状態)となります。暖機運転状態では実際の動作速度を指定された速度より下げて運転し、対象軸の動作時間の経過とともに徐々に指定された速度へ動作速度を戻します。速度を下げる割合を暖機運転オーバーライドといい、この値が 100% のとき、ロボットは指定された速度で動作します。工場出荷時のパラメータ設定では、暖機運転オーバーライドは対象軸の動作時間にあわせて図 5-6 のように値が変化します。

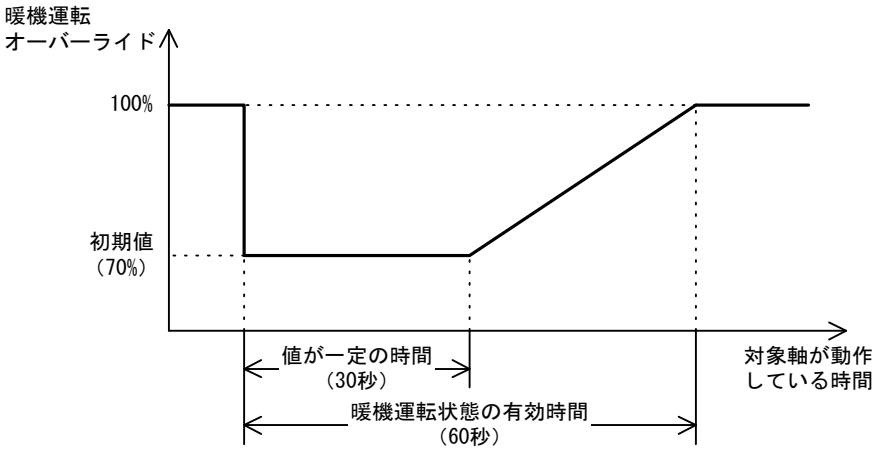


図 5-6：暖機運転オーバーライドの変化

⚠注意 暖機運転状態であっても、コントローラのモードが"MANUAL"となっている場合やジョグ動作、リアルタイム外部制御(Mxt 命令)による動作の場合は速度を下げず、本来の指定速度で動作します。

⚠注意 暖機運転状態では、本来の指定速度よりも低速で動作しますので、周辺装置とのインタロックを確実に実施するようにしてください。

⚠注意 対象軸の動作デューティが低いと、暖機運転モードを有効にしても誤差過大などのサーボエラーが発生することがあります。そのような場合は、プログラムを変更し速度・加減速度を下げてください。

また、暖機運転状態のときは、コントローラ前面の STATUS NUMBER をオーバーライド表示にすると左から 2 桁目にアンダースコア(_)が表示され、暖機運転状態であることを確認することができます。

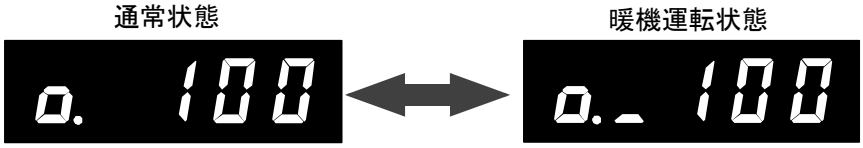


図 5-7：暖機運転状態のオーバーライド表示

対象軸が動作して暖機運転状態が解除されるとロボットは指定された速度で動作します。なお、解除後にロボットが停止し続けると低温下では再び関節部分が冷えてきます。そのため、対象軸が長時間(工場出荷時の設定は 60 分)停止し続けたときは、再び暖機運転状態となり速度を下げて動作します。

注 1) コントローラを電源遮断 / 再投入する場合、電源遮断の時間が短いと、ロボットの関節部分の温度はそれほど低下しません。したがって、暖機運転状態が解除された後に電源遮断 / 再投入するとき、電源遮断の時間が短い場合は暖機運転状態ではなく通常の状態で立ち上がります。

注 2) 対象軸とは、暖機運転モードで制御の対象となる関節軸のことでパラメータ WUPAXIS で指定した関節軸です。

(2) 機能の詳細

1) 暖機運転モードのパラメータ、専用入出力信号、状態変数

暖機運転モードでは、以下のパラメータ、専用入出力信号、状態変数が追加されています。これらの詳細については、408 ページの「5.1 動作パラメータ」、509 ページの「6.2 シーケンサリンク入出力機能」、99 ページの「4MELFA-BASIC V」を参照ください。

表 5-18：暖機運転モードのパラメーター覧

パラメータ名	説明と値
WUPENA	暖機運転モードの有効/無効を指定します。 0：無効 / 1：有効
WUPAXIS	暖機運転モードで制御の対象となる関節軸を、ビットの ON/OFF による 16 進数（下位ビットより J1,J2,…）で指定します。 ビット ON：対象軸 / ビット OFF：非対象軸
WUPTIME	暖機運転モードの処理で使用する時間（単位：分）を指定します。第 1 要素には有効時間、第 2 要素には再開時間を指定します。 有効時間：暖機運転状態となり速度を下げて動作する時間を指定します。(設定範囲：0～60) 再開時間：暖機運転状態が解除された後、対象軸が停止し続けた場合に、再び暖機運転状態となるまでの時間を指定します。(設定範囲：1～1440)
WUPOVRD	暖機運転状態における速度に関する設定をおこないます。第 1 要素には初期値、第 2 要素には値一定時間の割合を指定します。単位はともに%です。 初期値：暖機運転状態のときに動作速度に掛かるオーバーライド（暖機運転オーバーライド）の初期値を指定します。(設定範囲：50～100) 値一定時間の割合：暖機運転オーバーライドが初期値から変化しない時間を、有効時間に対する割合で指定します。(設定範囲：0～50)

表 5-19：暖機運転モードの専用入出力信号一覧

パラメータ名	区分	機能
MnWUPENA (n=1～3) (操作権必要)	入力	各メカの暖機運転モードを有効にします。(n：メカ番号)
	出力	暖機運転モードが有効となっていることを出力します。(n：メカ番号)
MnWUPMD (n=1～3)	出力	暖機運転状態であり速度を下げて動作することを出力します。(n：メカ番号)

表 5-20：暖機運転モードの状態変数

状態変数	機能
M_Wupov	暖機運転状態のとき、動作速度を下げるために指令速度に掛けるオーバーライド（暖機運転オーバーライド）の値を返します。
M_Wuprt	暖機運転状態を解除するために暖機運転モードの対象軸が動作しなければならない時間を返します。
M_Wupst	暖機運転状態が解除された後、再び暖機運転状態となるまでの時間を返します。

2) 暖機運転モードを使用するには

暖機運転モードを使用するには、パラメータで機能有効を指定します。また、専用入力信号で機能の有効 / 無効を切り替えることができます。

■パラメータで指定する

パラメータで暖機運転モードを有効とするには、パラメータ WUPENA に 1 を設定します。パラメータ変更後、コントローラの電源再投入により暖機運転モードが有効となります。ただし、次のような場合は、パラメータ WUPENA に 1 を設定しても暖機運転モードは有効にはなりません。

・パラメータ WUPAXIS に 0 が設定されている (暖機運転モードの対象軸が存在しない) 場合
・パラメータ WUPTIME の第 1 要素に 0 が設定されている (暖機運転状態である時間が 0 分) 場合
・パラメータ WUPOVRD の第 1 要素に 100 が設定されている (暖機運転状態であっても速度を下げない) 場合

暖機運転モードを使用する場合は、これらのパラメータを適切な設定値に変更してください。

■専用入力信号で切り替える

専用入力信号 MnWUPENA(n=1 ～ 3 : メカ番号) を割り付けることにより、コントローラの電源を再投入することなく暖機運転モードの有効 / 無効を切り替えることができます。また、現在の有効 / 無効の状態は、専用出力信号 MnWUPENA(n=1 ～ 3 : メカ番号) で確認することができます。

注 1) この専用入力信号を機能させるには、あらかじめ前述のパラメータ設定により暖機運転モードを有効としておく必要があります。
注 2) この専用入力信号は外部入出力の操作権が必要です。また、運転中やジョグ動作中は入力を受け付けません。
注 3) この専用入力信号により指定された有効 / 無効の状態は、外部入出力の操作権がなくなっても保持されます。

3) 暖機運転モードが有効のとき

暖機運転モードが有効のとき、コントローラの電源を投入すると暖機運転状態となります。
暖機運転状態では指定された速度に対して暖機運転オーバーライドを掛けることにより、実際の動作速度を下げて運転します。対象軸の動作時間の経過とともに徐々に指定された速度へ動作速度を戻し、暖機運転状態が解除されるとロボットは指定された速度で動作するようになります。

■電源投入直後の初期状態

暖機運転モードが有効のとき、コントローラの電源を投入すると暖機運転状態となります。
ただし、暖機運転状態が解除された後、コントローラの電源を遮断 / 再投入するとき、電源を遮断していた時間が短い場合は、ロボットの関節部分の温度は電源遮断時からそれほど低下していないので、暖機運転状態ではなく通常の状態で立ち上がります。具体的には、以下の条件を満足すると通常の状態で立ち上がります。

条件 : 暖機運転状態が解除してから電源投入時まで対象軸が停止し続けていた時間が、パラメータ WUPTIME の第 2 要素 (暖機運転状態の再開時間) に指定されている時間より短い場合は、通常の状態で立ち上がります。

なお、専用入力信号 MnWUPENA(n=1 ～ 3 : メカ番号) により暖機運転モードを有効に切り替えたときは、常に暖機運転状態となります。

■暖機運転状態であるのかどうかを確認する手段

現在暖機運転状態であるのか、それとも通常の状態であるのかは、次の 3 つの手段で確認できます。

・コントローラ前面の STATUS NUMBER により確認する。
STATUS NUMBER をオーバーライド表示にすることにより確認することができます。暖機運転状態では、左から 2 桁目にアンダースコア (_) が表示されます。

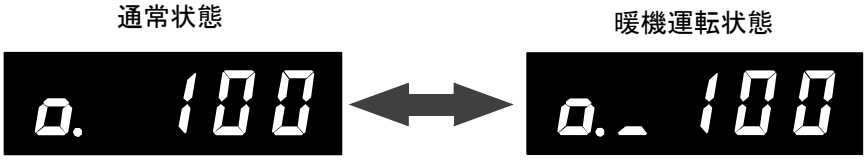


図 5-8 : 暖機運転状態のオーバーライド表示

・状態変数により確認する。
状態変数 M_Wupov(暖機運転オーバーライドの値) の値をモニタすることにより確認することがで

きます。通常の状態では M_Wupov の値は 100% となっていますが、暖機運転状態では 100% 未満の値となっています。

・専用出力信号により確認する。
暖機運転状態では、専用出力信号 MnWUPMD(n=1 ～ 3 : メカ番号) が出力されます。

■通常の状態 / 暖機運転状態の切り替え
暖機運転状態のとき、暖機運転モードの対象軸が動作を続け、その動作時間が暖機運転状態の有効時間を越えると、暖機運転状態は解除され通常の状態となります。その後、ロボットが停止し続けると低温下では関節部分が冷えてくるので、対象軸が長時間停止し続け、暖機運転状態の再開時間を越えた場合は、通常の状態から再び暖機運転状態に切り替わります。

・暖機運転状態の解除
対象軸の動作している時間の累積値が、暖機運転状態の有効時間を越えると、暖機運転状態は解除され通常の状態となります。暖機運転状態の有効時間は、パラメータ WUPTIME の第 1 要素に指定します。(工場出荷時の設定は 1 分)
対象軸が複数存在する場合は、すべての対象軸が有効時間を越えたときに暖機運転状態の解除となります。
なお、状態変数 M_Wuprt により、あとどのくらいの時間対象軸が動作すると暖機運転状態が解除するか確認することができます。
・通常の状態から暖機運転状態への切り替え
対象軸の停止し続けている時間が、暖機運転状態の再開時間を越えると、通常の状態から暖機運転状態へ切り替わります。暖機運転状態の再開時間は、パラメータ WUPTIME の第 2 要素に指定します。(工場出荷時の設定は 60 分)
対象軸が複数存在する場合は、その中の 1 軸でも暖機運転の再開時間を越えると暖機運転状態となります。
なお、状態変数 M_Wupst により、あとどのくらいの時間対象軸が停止し続けると暖機運転状態に切り替わるか確認することができます。
注) ロボットが動作していても対象軸が動作していない場合は、停止していると判断します。

通常の状態 / 暖機運転状態の切り替えの例をタイムチャートで表すと、図 5-9 のようになります。

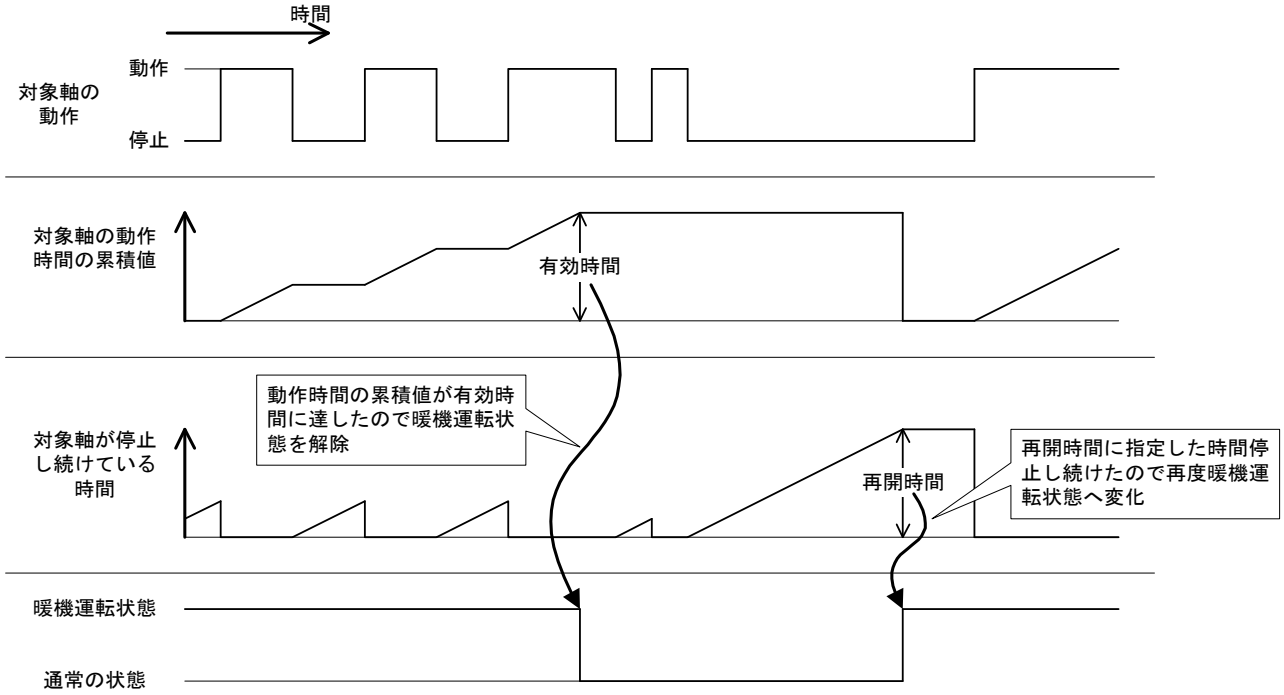


図 5-9 : 通常の状態 / 暖機運転状態の切り替えの例

■暖機運転オーバーライドの値について
暖機運転状態のとき、速度を下げるために動作速度に掛かるオーバーライドを暖機運転オーバーライドといいます。暖機運転オーバーライドは対象軸が動作している時間により下図のように変化し、ロボットの動作に即時反映されます。暖機運転オーバーライドの初期値と暖機運転状態の有効時間に対するオーバーライドが変化しない時間の割合は、パラメータ WUPOVRD で指定します。(工場出荷時の設定は、初期値が 70%、割合は 50%(=30 秒))
また、この値は状態変数 M_Wupov で確認することができます。

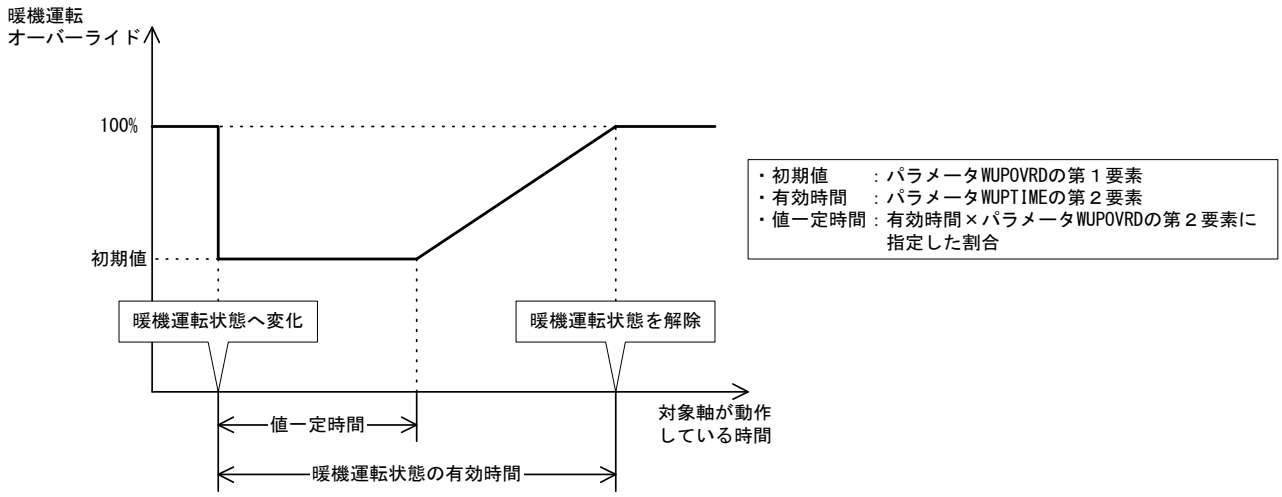


図 5-10：暖機運転オーバーライドの変化

なお、暖機運転状態における実際のオーバーライドは次のとおりです。
- 関節補間動作時 = (操作パネル (T/B) のオーバーライド設定値)x(プログラムオーバーライド (Ovrd 命令))×(関節オーバーライド (JOvrd 命令))× 暖機運転オーバーライド
- 直線補間動作時 = (操作パネル (T/B) のオーバーライド設定値)×(プログラムオーバーライド (Ovrd 命令))×(直線指定速度 (Spd 命令))× 暖機運転オーバーライド

注 1) コントローラのモードが "MANUAL" となっている場合やジョグ動作、リアルタイム外部制御 (Mxt 命令) による動作の場合は、暖機運転オーバーライドは反映されず、本来の指定速度で動作します。
注 2) 暖機運転状態では、本来の指定速度よりも低速で動作しますので、周辺装置とのインタロックを確実に実施するようにしてください。
注 3) 対象軸が複数存在する場合、暖機運転オーバーライドは対象軸中、最小の動作時間を使って算出されます。ある対象軸が動作せず、状態変数 M_Wuprt の値が変化しない場合は、他の対象軸がどれだけ動作しても暖機運転オーバーライドの値は変化しません。
また、各対象軸の動作 / 停止の状況によっては、値が 100% となる前に初期値に戻る場合もあります。
例えば、暖機運転オーバーライドが初期値より大きな値となっているときに、ある対象軸が通常の状態から暖機運転状態へと切り替わると、その軸の動作時間が最も小さく (動作時間は 0 秒) なるので、暖機運転オーバーライドは初期値に戻ります。

(3) アラームが発生する場合
1) 暖機運転状態で動作していても誤差過大のエラーが発生する
- 暖機運転オーバーライドが初期値のときにエラーとなる場合は、初期値の値 (パラメータ WUPOVRD の第 1 要素) を小さくしてください。
- 暖機運転オーバーライドが 100% に向かって増加している最中にエラーとなる場合は、暖機運転状態の有効時間もしくは値一定の時間が短すぎると思われます。パラメータ WUPTIME の第 1 要素 (有効時間) もしくは WUPOVRD の第 2 要素 (値一定の時間割合) の値を大きくしてください。
- 上記の対処をおこなってもエラーが解消されない場合は、動作プログラムを変更し、速度・加減速度を下げてください。

2) 暖機運転状態が解除されると誤差過大のエラーが発生する
- パラメータ WUPTIME の第 1 要素の値を大きくして、暖機運転状態の有効時間を長くしてください。

・ロボットの負荷や周囲温度が仕様範囲内か確認してください。
・暖機運転状態が解除された後、対象軸が長時間停止し続けていないか確認してください。そのような場合は、パラメータ WUPTIME の第 2 要素の値を小さくして、再び暖機運転状態となるまでの時間を短くしてください。
・上記の対処をおこなってもエラーが解消されない場合は、動作プログラムを変更し、速度・加減速度を下げてください。

3)いつまでたっても暖機運転状態が解除されない
・全く動作しない関節軸が暖機運転モードの対象軸となっていないかパラメータ WUPAXIS の設定値を確認してください。
・対象軸が暖機運転状態の再開時間(パラメータ WUPTIME の第 2 要素)よりも長い時間停止し続けていることがないか確認してください。
・指定されている速度が非常に低速(関節補間時のオーバーライドで概ね 3 ～ 5%)で動作し続けていないか確認してください。指定速度が低速の場合は、暖機運転モードを使用する必要はありませんので、暖機運転モードを無効にしてください。

5.20 特異点通過機能について

(1) 機能の概要

弊社のロボットは、直交座標系における位置データを使って直線補間動作の演算や教示位置の記憶などをおこなっています。例えば垂直 6 軸タイプのロボットの場合、X,Y,Z,A,B,C の座標値により位置データを表していますが、同じ位置データであってもロボットは複数の姿勢をとることができます。そこで、座標値と共に構造フラグ (姿勢を表すフラグ) を用いることにより、ロボットの姿勢を複数の中からひとつに定めています。

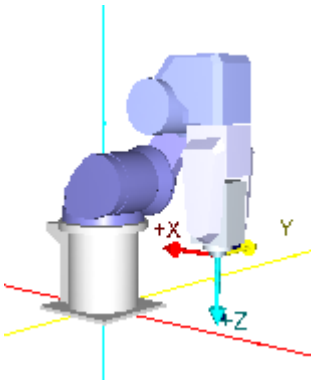
ところが構造フラグを用いても、このフラグが切り替わる位置では、ある関節軸のとり得る角度に無限の組み合わせがあるため、ロボットを所望の位置と姿勢に動作させることができません (例えば垂直 6 軸タイプのロボットの場合、J5 軸 = 0 度のとき J4 軸と J6 軸が一意に定まりません)。この位置のことを特異点といい、直交ジョグや直線補間などで通過することができませんでした。そのため、今までは作業エリアの中に特異点が存在しないようにレイアウトを工夫したり、どうしても特異点を通過する必要があるときは関節補間で動作させるなどの対応をとってきました。

特異点通過機能とは、直交ジョグや直線補間などで特異点を通過することを可能とする機能であり、これにより直線補間による作業エリアを拡大してレイアウトの自由度を高めることができます。

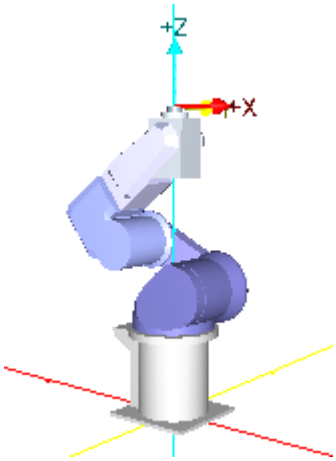
■通過可能な特異点の位置

特異点通過機能で通過することのできる特異点の位置は以下のとおりです。

垂直 6 軸タイプの場合

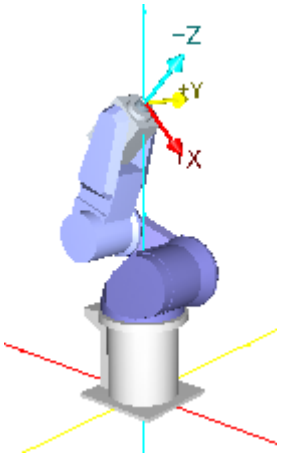


① J5 軸 = 0 度となる位置
構造フラグの NonFlip/Flip が切り替わる位置です。



② J1 軸の回転軸上に J5 軸中心が存在する位置
構造フラグの Right/Left が切り替わる位置です。

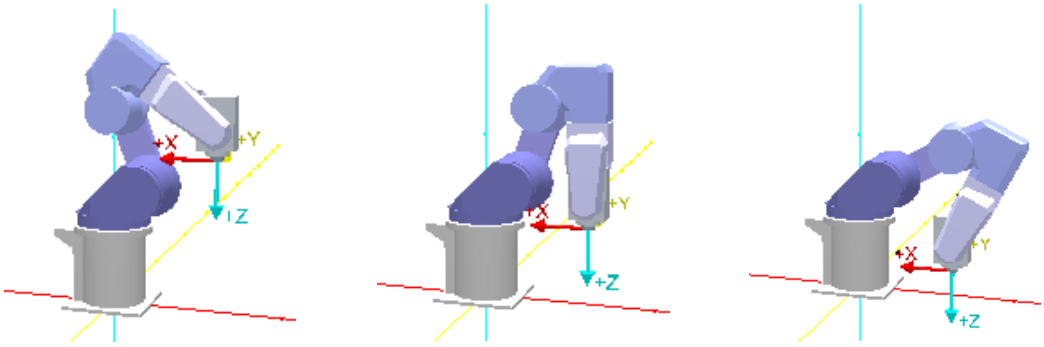
垂直 5 軸タイプの場合



① J1 軸の回転軸上にメカニカルインタフェース中心が存在する位置
構造フラグの Right/Left が切り替わる位置です。

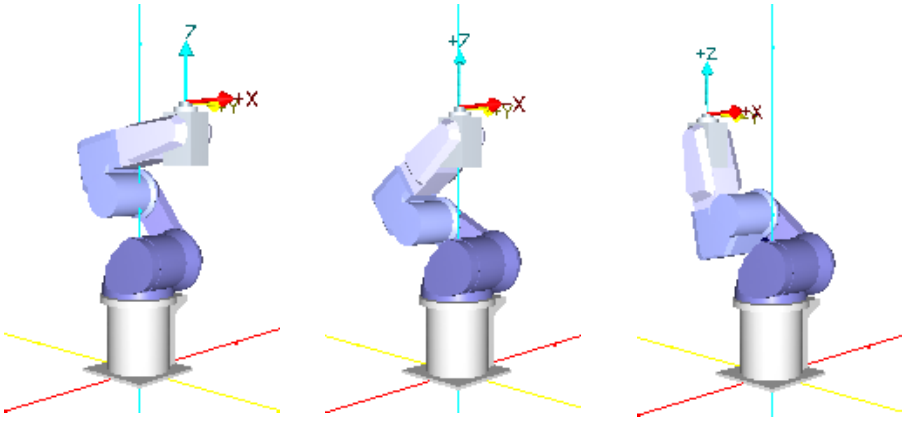
■特異点通過機能が有効のときの動作
特異点通過機能を有効にすると、直交ジョグや直線補間などで位置 A から位置 B(特異点位置) を経由して位置 C へ (その逆も) 動作できるようになります。このとき、位置 B の通過前後で構造フラグの値が切り替わります。
特異点通過機能が無効 (または対応していない) の場合は、位置 A から位置 B へ動作する前にエラーで止まります。直交ジョグの場合は、位置 B の直前で停止します。

位置 A　位置 B　位置 C



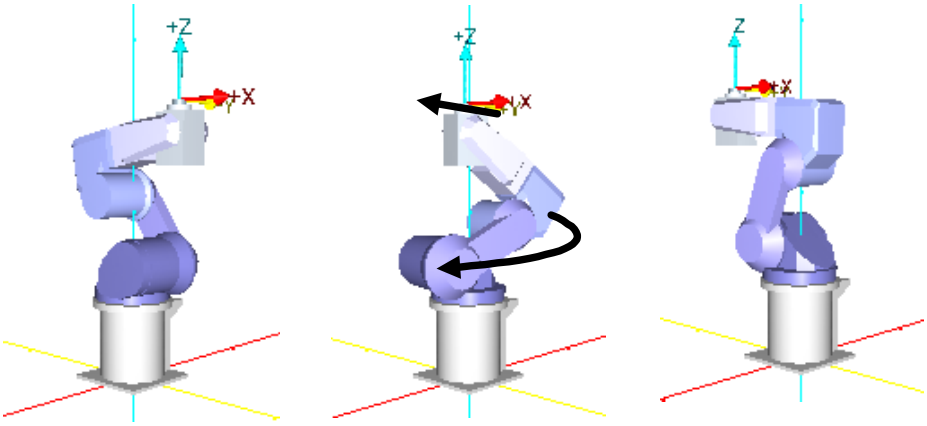
特異点を通過できるのは、ロボットの動作軌跡が特異点上を通過する場合です。動作軌跡が特異点上から外れている (特異点近傍を通過する) 場合は、構造フラグの値を切り替えずに動作します。
・位置 D → E → F : 特異点を通過する場合 (位置 E の前後で構造フラグが切り替わります。)

位置 D　位置 E　位置 F



・位置 G → H → I : 特異点近傍を通過する場合 (構造フラグは切り替わりません。)

位置 G　位置 H　位置 I



⚠注意　特異点近傍を通過する際、上図の位置 H のようにロボットが大きく回転する場合があります。位置の教示などロボットに近づいて作業する場合は、ロボットから目を離さず、接触しないよう十分ご注意ください。

■特異点通過機能を使用するには
特異点通過機能をジョグ操作で使用するには、パラメータ FSPJOGMD に 1(有効) を指定し、コントローラの電源を再投入します。自動運転で使用するには補間命令の TYPE 指定で定数 2 に 2 を指定します。

■対象機種
特異点通過機能に対応した機種は以下のとおりです。

対応機種
RV-2F、RV-4F シリーズ、RV-4FJL、RV-7F シリーズ、RV-7FLL、RV-13F シリーズ、RV-20F、RV-50F シリーズ

対象以外の機種で特異点通過機能を有効に設定しても、ジョグ操作では通常の動作（姿勢を維持したまま構造フラグを切り替えない動作）、自動運転ではエラーとなります。

■制限事項
特異点通過機能を使用するにあたり、以下の制限事項が存在します。
(1) 付加軸をマルチメカとして使用している場合、特異点通過機能は使用できません。
(2) ロボット付加軸の同期制御を使用している場合、特異点通過機能は使用できません。
(3) コンプライアンスモードが有効の場合、特異点通過機能は使用できません。
(4) 衝突検知機能が有効の場合、特異点通過機能は使用できません。
(5) メンテ予報の情報収集レベルはレベル 1(出荷時の設定) としてください。

(2) ジョグ動作の特異点通過機能
ジョグ動作の特異点通過機能は、パラメータ FSPJOGMD で有効 (1)/ 無効 (0) を指定します。

FSPJOGMD	直交ジョグ	ツールジョグ	ワークジョグ	三軸直交ジョグ	円筒ジョグ	関節ジョグ
0（出荷時設定値）	通常の直交ジョグ	通常のツールジョグ	通常のワークジョグ	設定値は影響しません。(XYZ は姿勢を維持したまま、構造フラグを切り替えません。)	設定値は影響しません。(姿勢を維持したまま、構造フラグを切り替えません。)	設定値は影響しません。
1	特異点通過の直交ジョグ	特異点通過のツールジョグ	特異点通過のワークジョグ			

①通常の直交 / ツール / ワークジョグとは、姿勢を維持したまま構造フラグを切り替えないジョグ動作のことです。
②特異点通過機能が使用できないロボットでは、パラメータ FSPJOGMD の設定値を変更しても通常の動作（姿勢を維持したまま構造フラグを切り替えない動作）となります。
特異点通過機能に対応した機種は RV-F シリーズです。(詳細は前述の「■対象機種」を参照)。
③特異点通過では複数軸同時にジョグ動作させることはできません。ある軸が動作しているときに他の軸を動作させようとしても、その動作は無視されます。
④ T/B を使ってジョグ動作をおこなっている際、特異点に近づくと特異点近傍警告が発生します。特異点近傍警告については、458 ページの「5.17 特異点近傍警告について」を参照してください。
⑤専用入力信号によるジョグ動作にもパラメータ FSPJOGMD の指定が反映されます。

(3) 位置データの確認操作 (ポジションジャンプ) の特異点通過機能
位置データの確認操作 (ポジションジャンプ) にもパラメータ FSPJOGMD の指定が反映されます。

FSPJOGMD	MO ポジション移動	MS ポジション移動
0 (出荷時設定値)	設定値は影響しません。 関節モードによるポジション移動を行います。	通常のポジション移動 (姿勢を維持したまま、構造フラグを切り替えません。)
1		特異点通過のポジション移動

⚠注意　T/B から補間命令 (例 : Mvs P1) をダイレクト実行した場合、パラメータ FSPJOGMD が 1(有効) となっていると、Type 指定で特異点通過を指定していなくても特異点通過有効で動作します。

(4) 自動運転の特異点通過機能
自動運転で特異点通過機能を使用するには、対象となる補間命令ごとに Type 指定で指定します。

Type (タイプ)

【機能】
補間命令の Type 指定で特異点通過機能を指定します。対応している補間命令は直線補間 (Mvs)、円弧補間 (Mvr, Mvr2, Mvr3) です。特異点通過の指定が可能な機種については 469 ページの「■対象機種」を参照してください。

【書式】

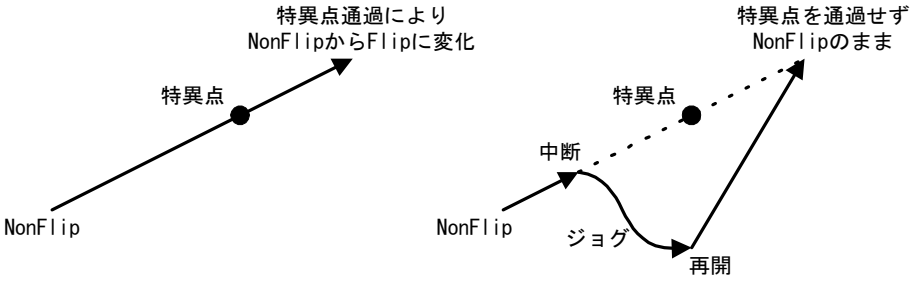
```
Type □ < 定数 1>, < 定数 2>
```

【用語】
<定数1>　0/1　:近回り/遠回り動作
<定数2>　0/1/2:等価回転/三軸直交/特異点通過

【文例】
```
1 Mvs P1 Type 0,2          ' 現在位置から P1 まで特異点通過有効で直線補間動作します
2 Mvr P1,P2,P3 Type 0,2    ' P1 ～ P3 まで特異点通過有効で円弧補間動作します
```

【解説】
(1) 特異点通過機能が使用できないロボットで < 定数 2> に 2 を指定したときは、実行時エラーとなります。
(2) 特異点通過を指定したときは、始点・終点間の構造フラグのチェックをおこないません。また、目的位置の構造フラグが特定できないので、動作開始前の目的位置・中間位置の動作範囲チェックをおこないません。
(3) Spd 命令で速度を指定したときは、上限を指定された速度とし、特異点近傍では速度エラーが発生しない速度まで自動的に低下させて動作します。
(4) 特異点通過を指定した補間命令には最適加減速が適用されず、固定加減速で動作します。このとき、Accel 命令の指定により加速時間と減速時間が異なる場合は、時間の長い方を加速、減速の両方に使用します。
(5) 特異点通過を指定した補間命令には Cnt 命令の指定が適用されず、加減速ありで動作します。
(6) 円弧で現在位置と始点の位置が異なるときは、Type 指定で特異点通過を指定していても、今までどおり三軸直交の直線補間で始点まで動作します。

(7) 特異点通過を指定した補間の動作を中断し、ジョグで移動後に動作を再開させると、パラメータ RETPATH にしたがって中断位置まで動作します。パラメータ RETPATH が 0(無効：中断位置へ戻らない)の場合、下図のように再開後の動作経路が特異点を通過しないと構造フラグが切り替わらず、補間完了時のロボットの姿勢が動作を中断しない場合と異なることがあります。



(8) 特異点通過を指定した場合、通常の直線補間などに比べて動作速度が遅くなることがあります。また、特異点通過機能は複雑な処理をおこなうため、プログラム実行速度に影響の出る場合があります。特異点通過の指定は、必要な補間命令のみおこなうようにしてください。

5.21 衝突検知機能について

(1) 機能の概要

ロボットを動作させて様々な作業をおこなうとき、作業者の操作ミスや動作プログラムの誤り等により、ロボットとワーク、周辺装置が干渉してしまう場合があります。そのような場合、従来はロボットのモーター駆動を制御しているサーボの保護機能（誤差過大検知等）でロボットを停止させることにより、ロボットのハンドやアーム部分、ワーク、周辺装置等の損傷軽減をはかっていました。ところが、ロボットの高速化・使用モーターの大容量化により、干渉時にかかる力がより大きなものになると、サーボの保護機能だけでは損傷を防ぐことが困難になります。

衝突検知機能は、従来のサーボの保護機能よりも高感度に干渉を検知し、より早くロボットを停止させることにより、損傷の軽減をはかることを目的とした機能です。

⚠警告　衝突検知機能を有効にしていても、動作中のロボットとの接触による作業者の負傷を防ぐことはできません。衝突検知機能の有効 / 無効にかかわらず、定められた安全ルールに則って作業をおこなってください。

⚠注意　衝突検知機能を有効にしていても、周辺装置との干渉によるロボット、ハンド、ワーク等の損傷を 100% 阻止することはできません。基本的には周辺装置に干渉しないよう十分に注意してロボットを操作、運用してください。

■干渉の検知原理

ロボットが周辺装置等に干渉すると、各関節軸の位置指令に対して実際の位置が追従しないため、サーボのフィードバック制御により、より大きなトルクが発生します。干渉が解消されなければ、発生するトルクはさらに増加することとなり、干渉していない場合とは大きく異なった値となります。

衝突検知機能では、このようなサーボの特性を利用して干渉を検知しています。まず、現在の位置指令や負荷設定等に基づいて各関節軸で必要となるトルクを推定します。次に、その値と実際に発生しているトルクを逐次比較します。そして、その差が許容範囲（検知レベル）を超えたときに干渉が起きたと判断し、サーボオフしてロボットを急停止させます。

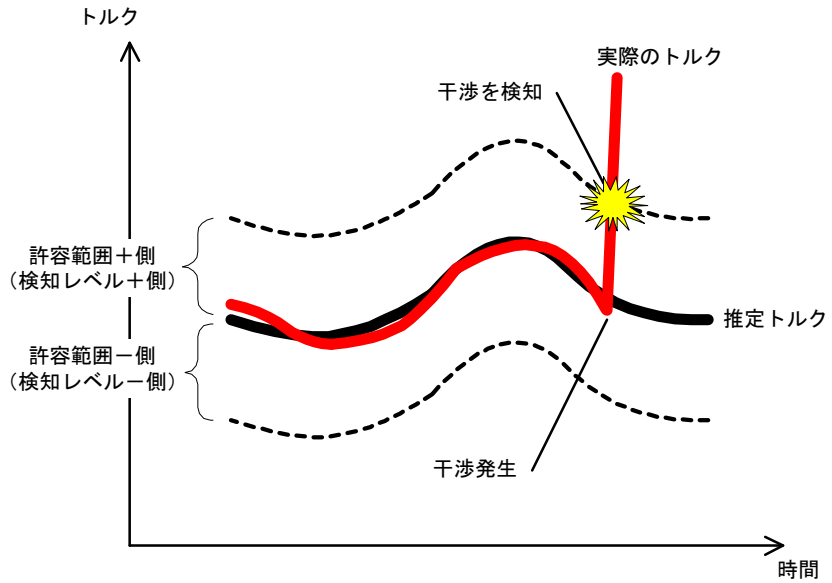


図 5-11：干渉の検知原理

(2) 関連パラメータ

衝突検知機能に関するパラメータには、以下のものがあります。これらの詳細については、408 ページの「5.1 動作パラメータ」、456 ページの「5.16 ハンド、ワーク条件設定 (最適加減速設定) について」をご参照願います。

表 5-21 : 衝突検知機能に関連するパラメータ

パラメータ名	説明と値	出荷時設定値
COL	衝突検知機能の使用可 / 不可と電源投入直後の有効 / 無効を定義します。 要素 1 : 衝突検知機能の使用可 (1)/ 不可 (0) を指定します。 要素 2 : プログラム運転時の初期状態を指定します。有効 (1)/ 無効 (0) 要素 3 : ジョグ操作時の有効 / 無効を指定します。有効 (1)/ 無効 (0)/NOERR モード (2)	RH-3FH/6FH/12FH/20FH シリーズ : 1,0,1 RV-F シリーズ : 0,0,1 RH-3FHR シリーズ : 1,1,1
COLLVL	プログラム運転時における各関節軸の検知レベル (検知の感度) の初期値を設定します。この値は、衝突検知機能の中で定めている検知レベル基準値に対する倍率となっています。値が小さいほど検知レベルは高くなります。 設定範囲 : 1 ～ 500　単位 : %	機種により異なります。
COLLVLJG	ジョグ操作時 (中断中を含む) における各関節軸の検知レベル (検知の感度) を設定します。この値は、衝突検知機能の中で定めている検知レベル基準値に対する倍率となっています。値が小さいほど検知レベルは高くなります。 設定範囲 : 1 ～ 500　単位 : %	機種により異なります。
HNDDAT* * は 0 ～ 8	ハンドの条件を設定します。(ツール座標系で設定します。) HNDDAT0 は電源投入直後の初期条件として採用されます。 (重量 , 大きさ X, 大きさ Y, 大きさ Z, 重心 X, 重心 Y, 重心 Z)　単位 : Kg, mm	機種により異なります。
WRKDAT* * は 0 ～ 8	ワークの条件を設定します。(ツール座標系で設定します。) WRKDAT0 は電源投入直後の初期条件として採用されます。 (重量 , 大きさ X, 大きさ Y, 大きさ Z, 重心 X, 重心 Y, 重心 Z)　単位 : Kg, mm	0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
HNDHOLD* * は 1 ～ 8	HOpen, HClose 命令を実行したときの、ワークの把持 (1)/ 未把持 (0) を指定します。 要素 1 : HOpen 命令実行時の状態を指定します。 要素 2 : HClose 命令実行時の状態を指定します。	0,1

(3) 衝突検知機能を使用するには

衝突検知機能を使用するには、はじめにパラメータ COL の要素 1 に「使用可 (1)」を指定し、コントローラの電源を再投入します。次に衝突検知機能の設定 (有効 / 無効の指定と検知レベル) を、ジョグ操作時とプログラム運転時に対してそれぞれ個別におこないます。

(473 ページの「表 5-21 : 衝突検知機能に関連するパラメータ」も合わせて参照願います。)

1) ジョグ操作時の使い方

ジョグ操作時の衝突検知機能は、すべてパラメータにより設定をおこないます。したがって、コントローラの電源投入中に有効 / 無効等設定を変更しても、電源を再投入するまでその変更は反映されません。設定をおこなうパラメータは表 5-22 のとおりです。

表 5-22 : ジョグ操作時の衝突検知機能において設定するパラメータ

パラメータ名	説明と値	出荷時設定値
COL	衝突検知機能の使用可 / 不可と電源投入直後の有効 / 無効を定義します。 要素 1 : 衝突検知機能の使用可 (1)/ 不可 (0) を 1 にします。 要素 3 : ジョグ操作時の有効 (1)/ 無効 (0) を指定します。有効 (1)/ 無効 (0)/NOERR モード (2)	RH-3FH/6FH/12FH/20FH シリーズ : 1,0,1 RV-F シリーズ : 0,0,1 RH-3FHR シリーズ : 1,1,1
COLLVLJG	ジョグ操作時 (中断中を含む) における各関節軸の検知レベル (検知の感度) を設定します。	機種により異なります。
HNDDAT0	ハンドの条件を設定します。(ツール座標系で設定します。)	機種により異なります。
WRKDAT0	ワークの条件を設定します。(ツール座標系で設定します。)	0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0

■衝突検知レベルの調整

ジョグ操作時の検知レベル (検知の感度) は低めに設定してあります。より高い検知レベルが必要な場合は、パラメータ COLLVLJG を調整してご使用ください。また、トルクの推定を精度よくおこなうため、パラメータ HNDDAT0, WRKDAT0 も正しく設定してください。

◇◆◇ ポイント ◇◆◇

検知レベルを上げすぎる (設定値を小さくしすぎる) と、ロボットの位置と姿勢によっては、干渉を誤検知する場合があります。そのような場合は、検知レベルを下げて (設定値を大きくして) ご使用ください。

コントローラのソフトウェアバージョン R6b 版 /S6b 版以上、かつ RT ToolBox2 Ver. 3.40S 以上をご使用の場合、RT ToolBox2 のオシログラフ機能で「衝突検知レベル参考値」を選択することができます。これにより、パラメータ COLLVLJG 設定値の調整作業の容易化が図れます。詳細は 477 ページの「3) 補足事項」を参照ください。

■干渉を検知したときの挙動

ジョグ操作中に周辺装置等との干渉を検知すると、1010 番台 (1 桁目は軸番号) のエラーが発生し、ロボットはサーボオフして停止します。なお、NOERR モードの場合 (パラメータ COL の要素 3 に 2 を設定した場合)、エラーは発生せず、サーボオフして停止します。(ただし、エラー履歴には 1010 番台のエラーを残します。)

■干渉後の操作

ハンドやアームが周辺装置等と干渉した状態のままサーボオンすると、再度衝突検知状態となりサーボオンできない場合があります。サーボオンを繰り返してもエラーとなる場合は、一旦ブレーキを解除してアームを逃がしてからジョグ操作をおこない、干渉状態を解消してください。

■ ジョグ操作中に衝突検知を一時的に無効にする方法

T/B の「 RESET 」キーを押しながらサーボオン及びジョグ操作をおこなうと、「 RESET 」キーを押している間は衝突検知が無効となります。

2) プログラム運転時の使い方

プログラム運転時の衝突検知機能は、パラメータで初期状態を設定しますが、実際には MELFA-BASIC V の命令を使ってプログラムの中で設定を変更して使用します。初期状態を設定するパラメータと衝突検知機能に関連する命令は、それぞれ下表のとおりです。なお、命令の詳細については、164 ページの「4.13 命令の詳細説明」、288 ページの「4.14 ロボット（システム）状態変数の詳細解説」をご参照願います。

表 5-23：プログラム運転時の衝突検知機能において設定するパラメータ

パラメータ名	説明と値	出荷時設定値
COL	衝突検知機能の使用可 / 不可と電源投入直後の有効 / 無効を定義します。 要素 1：衝突検知機能の使用可 (1)/ 不可 (0) を 1 にします。 要素 2：プログラム運転時の初期状態の有効 (1)/ 無効 (0) を 1 にします。	RH-3FH/6FH/12FH/20FH シリーズ：1,0,1 RV-F シリーズ：0,0,1 RH-3FHR シリーズ：1,1,1
COLLVL	プログラム運転時における各関節軸の検知レベル（検知の感度）の初期値を設定します。	機種により異なります。
HNDDAT* ＊は 0 ～ 8	ハンドの条件を設定します。（ツール座標系で設定します。）	機種により異なります。
WRKDAT* ＊は 0 ～ 8	ワークの条件を設定します。（ツール座標系で設定します。）	0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
HNDHOLD* ＊は 1 ～ 8	HOpen, HClose 命令を実行したときの、ワークの把持 (1)/ 未把持 (0) を指定します。	0,1

表 5-24：プログラム運転時の衝突検知機能で使用する MELFA-BASIC V 命令と状態変数

命令 / 状態変数	説明
ColChk	突検知機能の有効 / 無効 /NOERR モードを指定します。 例：ColChk On　　' 衝突検知機能を有効にする。
ColLvl	衝突検知機能の検知レベル（検知の感度）を関節軸毎に設定します。この値はパラメータ ColLvl 同様、衝突検知機能の中で定めている検知レベル基準値に対する倍率となっています。（単位：%） 例：ColLvl 80, 120, 120, 120, 50, 80,　　'J1 ～ J6 軸の検知レベルを設定
LoadSet	ハンド、ワークの条件を指定します。プログラム運転中に使用するハンドや把持するワークが変更となる場合に使用します。 例：LoadSet 1, 0　　' パラメータ HNDDAT1、WRKDAT0 の条件を指定
J_ColMxl	推定トルクと実際のトルクの差の最大値を検知レベルに変換して返します。ColLvl 命令の引数の調整時に参照します。（単位：%）
M_ColSts	干渉を検知したときに 1 を返します。NOERR モードの割込み条件に使用します。
P_ColDir	干渉を検知したときのロボットの動作方向 (X,Y,Z の動作比率) を返します。NOERR モードにおける退避動作で使用します。

◇◆◇ ポイント ◇◆◇

プログラム全体に渡って衝突検知機能を有効にすると、それに伴い誤検知を起こす可能性も高くなります。誤検知をなくすには検知レベルを下げなければならないため、その結果、衝突検知が必要な動作に対して干渉の検知感度が低くなってしまいます。検知感度を高くするためにも、干渉の可能性のある動作に対してのみ衝突検知機能を使用するようにしてください。

◇◆◇ ポイント ◇◆◇

衝突検知機能を有効にすると、プログラムによっては実行時間（タクトタイム）が延びる場合があります。タクトタイムへの影響を軽減するために、プログラム全体にわたって衝突検知機能を有効にするのではなく、干渉の可能性のある動作に対してのみ使用するようにしてください。

■衝突検知レベルの調整
ロボットの動作に応じてプログラム運転時の検知レベル（検知の感度）を調整してください。参考までに、調整手順の一例を以下に示します。なお、トルクの推定を精度よくおこなうため、ワーク条件とハンド条件も正しく設定してください。

表 5-25：プログラム運転時の検知レベル調整手順の一例

手順	内容
1	衝突検知機能を使用する動作の前後に、ColLvl 命令と ColChk 命令を追加します。
2	干渉を誤検知しないよう、検知レベルを低く（ColLvl 命令の引数に 300 など大きな値を）設定します。
3	プログラムを運転し、対象となる動作における J_ColMxl の値をモニタします。なお、値がばらつくことがありますので、対象となる動作を複数回繰り返し、その都度 J_ColMxl の値を控えます。
4	複数の J_ColMxl の値から各関節軸における最大値を求め、その値にマージン（例えば 20%）を加えたものを ColLvl 命令の引数とします。
5	手順 4 で求めた値を ColLvl 命令に設定してプログラムを運転し、衝突検知機能を使用する動作で誤検知しないことを確認します。誤検知する場合は、ColLvl 命令の引数の値を徐々に大きくしていき、誤検知しなくなるまで検知レベルを下げていきます。

◇◆◇ ポイント ◇◆◇
動作速度を変更すると検知レベルの変更が必要となる場合があります。検知レベルを調整する際は、ロボットを実際に運用する速度で動作させて調整してください。

◇◆◇ ポイント ◇◆◇
複数台のロボットで衝突検知機能を使用する場合、モーター特性のばらつき等によるロボットの個体差や使用環境により、同一の動作であっても検知レベルの調整が必要となることがあります。また、ロボットの機種が異なる場合は、ロボット毎に検知レベルの調整が必要となります。

コントローラのソフトウェアバージョン R6b 版 /S6b 版以上、かつ RT ToolBox2 Ver. 3.40S 以上をご使用の場合、RT ToolBox2 のオシログラフ機能で「衝突検知レベル参考値」を選択することができます。これにより、ColLvl 命令設定値の調整作業の容易化が図れます。詳細は 477 ページの「3) 補足事項」を参照ください。

■干渉を検知したときの挙動
プログラム運転中に周辺装置等との干渉を検知すると、1010 番台（1 桁目は軸番号）のエラーが発生し、ロボットはサーボオフして停止します。なお、NOERR モードの場合、エラーは発生せず、サーボオフして停止します。（ただし、エラー履歴には 1010 番台のエラーを残します。）

■プログラム例
干渉を検知すると割込み処理により退避動作をおこないます。

```
1 Def Act 1,M_ColSts(1)=1 GoTo *HOME,S   ' 干渉を検知したときに実行する処理を割込みで定義
2 Act 1=1
3 ColLvl 80,120,120,100,80,80,,          ' 検知レベルを設定
4 ColChk On,NOErr                        ' NOERR モードで衝突検知機能を有効にする
5 Mov P1
6 Mov P2                                 ' 5 ～ 8 行実行中に干渉を検知すると割込み処理へ
                                           ジャンプする
7 Mov P3
8 Mov P4
9 ColChk Off                             ' 衝突検知機能を無効にする
10 Act 1=0
    :
11 *HOME                                 ' 干渉を検知したときの割込処理
12 ColChk Off                            ' 衝突検知機能を無効にする
13 Servo On                              ' サーボオンする
14 PESC=P_ColDir(1)*(-5)                 ' 退避動作の移動量（約 5mm の反転動作）を算出
15 PDST=P_Fbc(1)+PESC                    ' 退避位置を作成する
16 Mvs PDST                              ' 退避位置へ移動
17 Error 9100                            ' ユーザー定義の L レベルエラーを発生させて運転を中断
    :
```

3) 補足事項

■衝突検知機能では、現在の位置指令や負荷設定等に基づいて各関節軸で必要となるトルクを推定し、その値と実際に発生しているトルクを比較することで衝突判定をします。
実際に衝突が発生しなくても、通常運転中にアームに外力がかかる場合は衝突とみなす場合があります。たとえば、ハンド姿勢変更時にハンドが配管、配線により引っ張られ、ハンドに力がかかる場合には、その力の大きさによっては衝突と判断します。衝突検知有効時のロボット動作において、衝突以外の力が働かないか注意してください。

■ジョグ操作時とプログラム運転時の区別について
ジョグ操作時とプログラム運転時では、ロボットの動作速度や作業内容が大きく異なるので、それぞれに対して衝突検知機能を最適に働かせるために、設定は独立したものとなっています。ここで、″ジョグ操作時″と″プログラム運転時″とは、次の状態を指しています。

・ジョグ操作時　　：ジョグ動作中、自動運転中断中
・プログラム運転時：自動運転中、ステップ送り / 戻し動作中、位置データの確認動作中

これらは、図 5-12 のような状態遷移をとります。

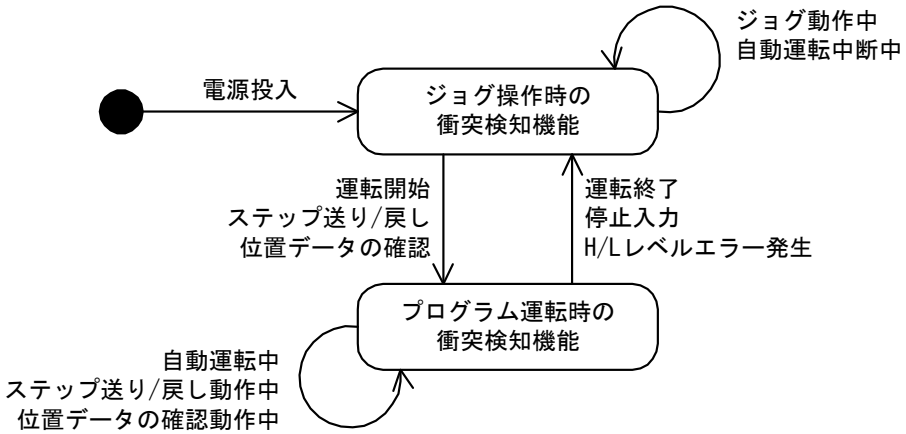


図 5-12：プログラム運転時 / ジョグ操作時の切り替えに関する状態遷移

したがって、例えばジョグ操作時の衝突検知機能が有効の場合、プログラム運転の中で衝突検知を無効に設定しても、停止ボタンを押して運転が中断するとジョグ操作時の設定に切り替わり、衝突検知は有効となります。

■サーボオフ中の衝突検知機能
ジョグ操作時、プログラム運転時ともサーボオフ中は、衝突検知機能は一時的に無効となります。

■オシログラフ機能の「衝突検知レベル参考値」について
コントローラのソフトウェアバージョン R6b 版 /S6b 版以上、かつ RT ToolBox2 Ver. 3.40S 以上をご使用の場合、RT ToolBox2 のオシログラフ機能のリアルタイムモニタで「衝突検知レベル参考値」を選択することができます。
「衝突検知レベル参考値」は、推定トルクと実トルクの差を衝突検知レベル (ColLvl 命令やパラメータ COLLVL、COLLVLJG の設定値) で表したもので、設定値を決める際の目安となります。

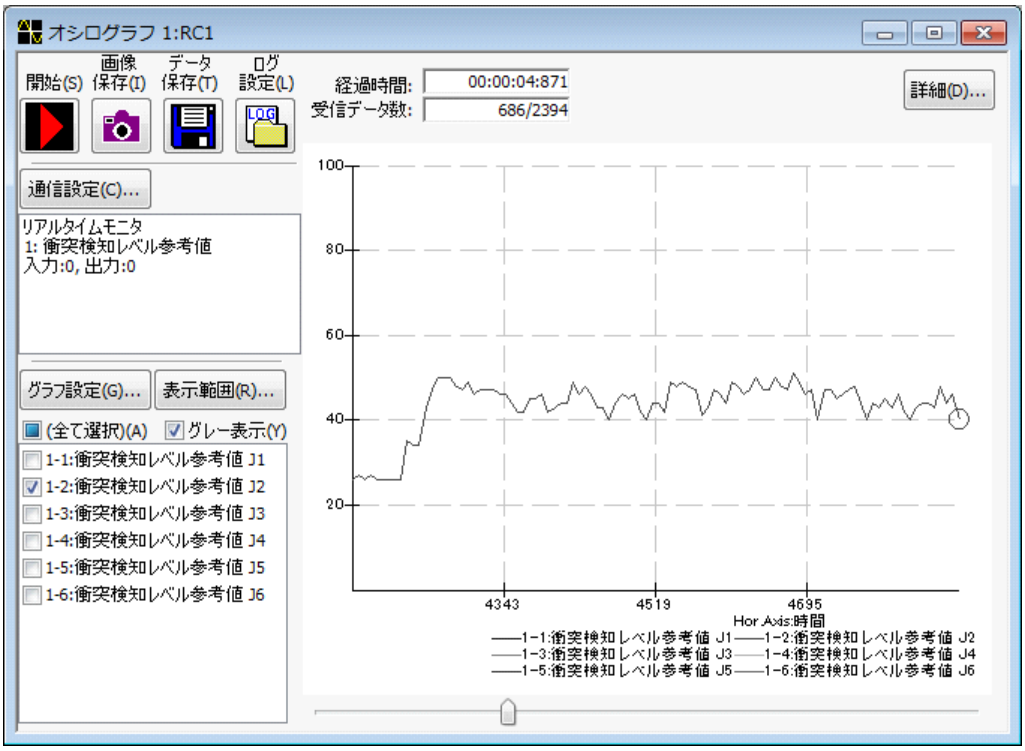


図 5-13：「衝突検知レベル参考値」のオシログラフ表示

衝突検知機能が有効、かつサーボ ON しているとき、図 5-13 に示すように衝突検知レベルがオシログラフへリアルタイムに出力されます。衝突検知を有効とする動作に対し、この値を参考にして衝突検知レベルを設定します。
以下に調整手順の一例を示します。なお、トルク推定を精度良く行うため、ワーク条件とハンド条件を正しく設定してください。

表 5-26：プログラム運転時の衝突検知レベル調整手順の一例

手順	内容
1	衝突検知機能を使用する動作の前後に、ColLvl 命令と ColChk 命令を追加します。
2	干渉を誤検知しないよう、検知レベルを低く（ColLvl 命令の引数に 300 など大きな値を ）設定します。
3	プログラムを運転し、対象となる動作における「衝突検知レベル参考値」をオシログラフでモニタします。なお、値がばらつくことがありますので、対象となる動作を複数回繰り返し、各軸の最大値を控えます。
4	手順 3 で求めた各軸の最大値にマージン（たとえば 20%）を加えたものを ColLvl 命令の引数とします。
5	手順 4 で求めた値を ColLvl 命令に設定してプログラムを運転し、衝突検知機能を使用する動作で誤検知しないことを確認します。誤検知する場合は、ColLvl 命令の引数の値を徐々に大きくしていき、誤検知しなくなるまで検知レベルを下げていきます。

5.22 過負荷レベルの最適化について

ご使用になるロボットの実際の周辺温度が 40 ℃以下の場合、実際の周辺温度をパラメータ：OLTMX に設定することで、ロボット動作時に働く過負荷エラーの検知レベル（モータの発熱を保護する機能）を、お客様の使用環境に合わせて最適化させ、連続動作性をより向上させることができます。

【対象機種】：RH-F シリーズ /RV-F シリーズ

表 5-27 に、パラメータ：OLTMX の説明を示します。
ロボットの周辺温度が 40 ℃以下に管理されている場合は、実際の周辺温度を本パラメータに設定してロボットを有効にご活用ください。

表 5-27：関連するパラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
過負荷検知ﾚﾍﾞﾙの最適化	OLTMX	整数 1	ロボット使用環境での周囲温度の上限値を設定します。 この設定値に基づきロボット動作時の過負荷検知レベルを最適化します。（単位：℃） 設定範囲：0 ～ 40	RH-3FH/6FH/12FH/20FH:40 RV-2Fｼﾘｰｽﾞ:40 RV-4F/7F/13F/20F/50Fｼﾘｰｽﾞ、RH-3FHR:30

注）周辺温度設定の注意

通年を通してロボット周辺温度が変化する場合は、その最大温度を設定してください。実際の温度より低い温度を設定すると、過負荷エラーが発生する前に、モータオーバーヒートエラーが発生する場合があります。
周辺温度パラメータ変更による連続動作性の改善効果は、機種毎・動作軸毎に異なります。
パラメータを変更しても、連続動作性能が変化しない場合があります。

5.23 パレット定義命令の多回転制限

J1 軸または J4 軸が ±180 度を超えることができるロボット機種 (RV-2F など) において、パレット定義命令 : DEFPLT を使用する場合は、J1 軸または J4 軸の関節角度が ±180 度をまたぐようなパレットは指定できません、もし、多回転をまたぐような位置を指定した場合は、パレット定義命令実行時にエラーが発生します。パラメータを変更することで実行できるようになります。

表 5-28 : 関連するパラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
パレット仕様	PLTSPEC	整数 1	パレット指定の各点について、多回転位置のエラーチェックを有効にする。 0 : 多回転位置のエラーチェックをしない 1 : 多回転位置のエラーチェックをする 1(多回転位置のエラーチェックをする) に設定した場合、RV-2F ロボットのように J1 軸や J4 軸の関節角度が ±180 度を超えて多回転する機種においては、多回転位置をまたぐ位置を指定した場合や、J1 軸や J4 軸が大きく回る位置を指定した場合は、パレット定義コマンドを実行したときにエラーとなります。 0(多回転位置のエラーチェックをしない) に設定すると、エラーが出なくなりますが、移動命令の近回り (TYPE) や、多回転フラグ (FL2) の指定において、多回転を考慮したプログラムを作成する必要があります。指定を誤ると関節軸が 1 回転して周辺装置などへ干渉する場合があります。必ずパレット全点に対して、低速での動作確認をおこなってください。	1:RV-Fシリーズ 0:その他機種

ハンド先端軸 (RV 機種では J6 軸、RH 機種では J4 軸) に加え、旋回軸の J1 軸および RV 機種の前腕ツイスト軸である J4 が ±180 度を超えて多回転することが可能です。

パレット定義命令では、これらの軸の多回転をまたがっての動作はできません。多回転フラグが異なる位置をパレット定義命令で指定して動作させると、遠回り指定の関節補間では J1 軸が回って反対に動作します。場合によっては思いもよらない動作となるため、パレット定義命令において下記の制限があります。

1) 始点の多回転フラグと、終点 A または終点 B または対角点の多回転フラグが異なるケース
多回転軸 (RV の J6 軸機種の場合は J1 軸および J4 軸、RV の 5 軸機種と RH 機種の場合は J1 軸) の多回転フラグについてパレット定義の各点で比較確認し、1 つでも多回転フラグが異なる場合はパレット定義命令の実行時にエラー L3750 が発生してエラー停止します。通常パレットと円弧パレットの両方を対象とします。

2) 1) の多回転フラグが一致していても関節角度が大きく異なるケース
多回転軸 (RV の J6 軸機種の場合は J1 軸および J4 軸、RV の 5 軸機種と RH 機種の場合は J1 軸) の関節座標位置についてパレット定義の各点と始点とを比較して、関節角度が ±180 度以上異なる場合はパレット定義命令の実行時にエラー L3760 が発生してエラー停止します。通常パレットと円弧パレットの両方を対象とします。

多回転をチェックする軸としては、ロボットの軸 (旋回軸である J1 軸、RV 機種の前腕ツイスト軸である J4 軸) を対象としており、付加軸およびユーザーメカの多回転については、上記チェックをしません。

5.24 干渉回避機能について

本機能は干渉チェックを行いながらロボットを動作させる機能です。干渉チェック対象として①ロボット同士（CR7xx-Q コントローラのみ）と②ロボットと自由平面リミット間があります。
ジョグ操作や自動運転時にロボット同士や自由平面リミットとの干渉を事前検知して停止させることにより、ロボットや周辺装置へのダメージを軽減させることができます。
干渉を事前検知すると、ロボットの動作は停止します。プログラムでの指定でアラームを発生させたり、回復動作をさせることができます。

【対象機種】：ロボット同士の干渉チェックは RH-F/RV-F シリーズの CR7xx-Q コントローラのみ。（ユーザメカは未対応）
ロボットと自由平面リミットとの干渉チェックは RH-F/RV-F シリーズ。（ユーザメカは未対応）
※ 対応しているコントローラのソフトウェアバージョンは 483 ページの「5.24.2 機器の準備と接続」を参照ください。

回避できる干渉の例
本機能を利用することで以下のようなロボット動作における干渉を未然に防止する制御が働きます。
1) ジョグ操作時の誤操作によりロボットをぶつけた。
2) 自動運転におけるプログラムミスでロボットをぶつけた。
3) 復旧時に意図した動作と違うシーケンスに入りロボット同士や周辺装置が衝突した。
4) 自動運転で初めて高速で動作させた時に、インターロックの不備で衝突した。
5) 2 台のロボットをオーバーライドが違う状態で動作させたためロボット同士が衝突した。

図 5-14：ロボット動作時の干渉チェック（自動運転・ジョグ運転）

⚠注意

本干渉回避機能は、ロボットアーム、ハンド、ワークをモデル化しておき、そのモデル同士が重なると干渉の危険が発生したと判断し、実際に干渉が起こらないようロボットの動作を制御する機能です。
ロボットの動作に応じた適切なモデル登録が必要になることから、干渉の防止を完全に保障するものではないことに注意してご利用願います。

5.24.1 操作手順

以下に干渉回避機能を使用するための操作手順概要を示します。

(1) ロボット同士の干渉チェック(CR7xx-Q コントローラのみ)

＜操作の流れ＞

(1) 機器の準備と接続 … 対象とするロボット 2 台、または 3 台と、RT ToolBox2 をインストールしたパソコンを用意します。
下記 483 ページの「5.24.2 機器の準備と接続」に詳細を示します。

↓

(2) 干渉チェックモデルの登録 … 干渉回避チェックの対象となるモデルをロボット本体、ハンド、ワークについて、ロボット本体を基準とした位置、大きさをパラメータで設定します。
484 ページの「5.24.3 干渉チェック用モデルの登録」に詳細を示します。

↓

(3) 付加軸の同期設定 … 走行軸などの直動軸を付加軸としてご使用の場合、その同期制御パラメータを設定します。
493 ページの「5.24.5 付加軸の対応」に詳細を示します。

↓

(4) ロボット CPU 間直接通信の設定 … パラメータで共有メモリ拡張機能を設定します。
(本機能は、ロボット CPU 間通信で共有メモリを占有します。)
494 ページの「5.24.6 共有メモリ拡張機能の設定(ロボット同士の干渉チェックの場合のみ)」に設定方法を示します。

↓

(5) ロボット間キャリブレーションの実施とチェック … 複数ロボット相互の位置関係の設定と、その確認をおこないます。
496 ページの「5.24.7 ロボット間キャリブレーション(ロボット同士の干渉チェックの場合のみ)」に詳細を示します。

↓

(6) 干渉回避機能の設定 … 干渉回避機能自体をパラメータで使用可能状態に設定して使います。
498 ページの「5.24.8 干渉回避機能の有効/無効設定」に詳細を示します。

↓

(7) 干渉回避機能の使用 … ジョグ操作時、およびプログラム実行時の干渉回避動作を 499 ページの「5.24.9 干渉回避機能の使用」以降に説明します。
また、500 ページの「5.24.10 サンプルプログラム」にサンプルプログラムを示します。

(2) 自由平面リミットとの干渉チェック

＜操作の流れ＞

(1) 機器の準備と接続 … 対象とするロボット 2 台、または 3 台と、RT ToolBox2 をインストールしたパソコンを用意します。
下記 483 ページの「5.24.2 機器の準備と接続」に詳細を示します。

↓

(2) 干渉チェックモデルの登録 … 干渉回避チェックの対象となるモデルをロボット本体、ハンド、ワークについて、ロボット本体を基準とした位置、大きさをパラメータで設定します。
484 ページの「5.24.3 干渉チェック用モデルの登録」に詳細を示します。

↓

(3) 付加軸の同期設定 … 走行軸などの直動軸を付加軸としてご使用の場合、その同期制御パラメータを設定します。
493 ページの「5.24.5 付加軸の対応」に詳細を示します。

↓

(5) 干渉回避機能の設定 … 干渉回避機能自体をパラメータで使用可能状態に設定して使います。
498 ページの「5.24.8 干渉回避機能の有効/無効設定」に詳細を示します。

↓

(6) 干渉回避機能の使用 … ジョグ操作時、およびプログラム実行時の干渉回避動作を 499 ページの「5.24.9 干渉回避機能の使用」以降に説明します。
また、500 ページの「5.24.10 サンプルプログラム」にサンプルプログラムを示します。

5.24.2 機器の準備と接続

表 5-29 に本機能を使用するための必要な機器を、図 5-15 に接続例を示します。
図を参考に必要な機器を接続してください。

(1) ロボット同士の干渉チェック (CR750-Q/CR751-Q コントローラのみ)

表 5-29 : 必要な機器

No.	機器	備考
1	ロボット (CR750-Q/CR751-Q コントローラ) 2 または 3 台 (最大)	本機能は iQPlatform の共有メモリを介したロボット CPU 間直接通信を利用します。 コントローラのソフトウェアバージョン : R3 版以上 (モデル形状 : 球) R3m 版以上 (モデル形状 : 球、円筒) 注) 同じソフトウェアバージョン同士で使用願います。
2	RT ToolBox2 をインストールしたパソコン	ロボット CPU と接続します。 RTToolBox2 のソフトウェアバージョン : 2.1L 版以上。

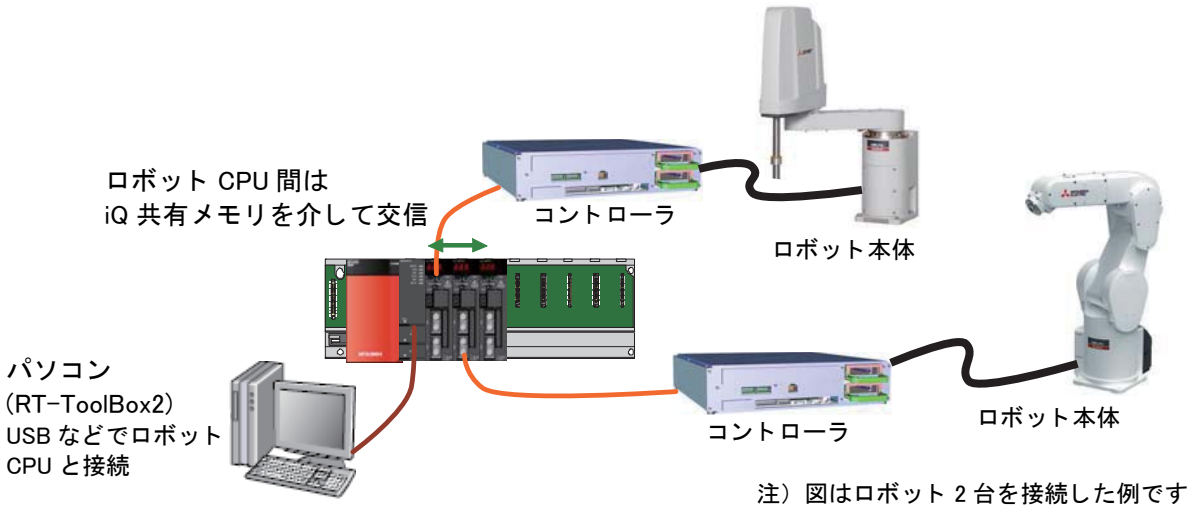


図 5-15 : 機器の接続

(2) 自由平面リミットとの干渉チェック

表 5-30 : 必要な機器

No.	機器	備考
1	ロボット 1 台	自号機に設定されている自由平面リミットのみ干渉チェックを行います。 コントローラのソフトウェアバージョン : R6b/S6b 版以上
2	RT ToolBox2 をインストールしたパソコン	ロボット CPU と接続します。 RT ToolBox2 のソフトウェアバージョン : 3.40G 版以上

<CR750-D/CR751-D コントローラ >

図 5-16 : 機器の接続 (CR750-D/CR751-D コントローラ)

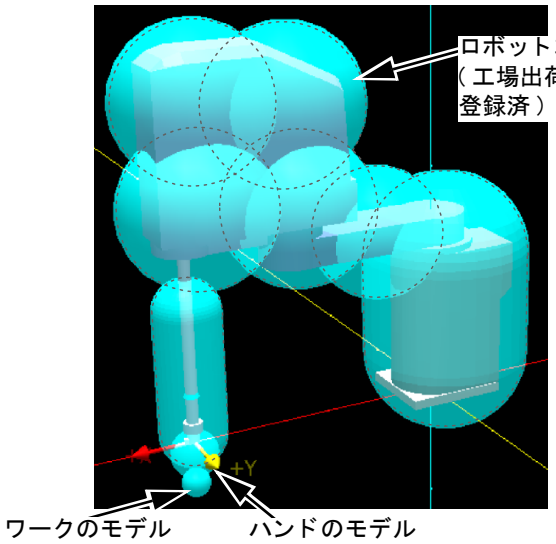
<CR750-Q/CR751-Q コントローラ>

図 5-17：機器の接続 (CR750-Q/CR751-Q コントローラ)

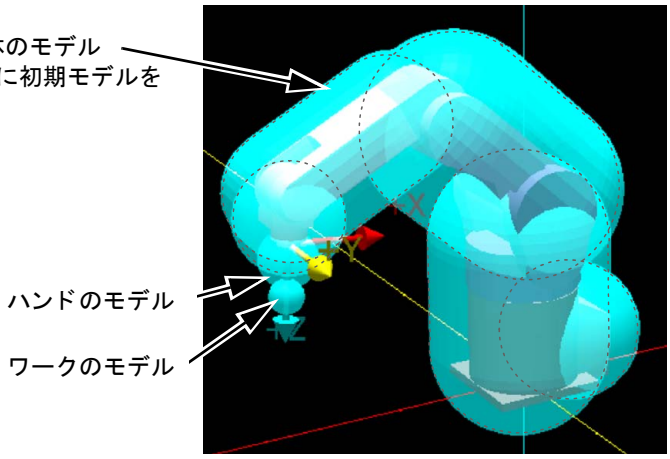
5.24.3 干渉チェック用モデルの登録

干渉チェックに使用するモデル (以下 " モデル ") をロボット本体を基準に登録します。

<RH-F シリーズ>



<RV-F シリーズ>



※ 図は RT ToolBox2 の画面イメージ (設定例) です。
球、および円筒の形がモデルを表します。
ハンド、およびワークのモデルはお客様で設定します。

図 5-18：モデルの登録例

必要な登録内容を表 5-31 にまとめます。各モデルはロボット本体、ハンド、ワークにつきそれぞれ 8 個まで登録できます。

表 5-31：モデルの登録内容

モデルの設定項目	モデルの種類		
	ロボット本体のモデル[注 1)]	ハンドのモデル	ワークのモデル
関係付け ・ロボット本体： 各モデルをロボット本体のどの部分に登録するかを設定 ・ハンド、ワーク： 各モデルに番号を設定 (ハンド番号、ワーク番号)	ロボット Jn 軸として登録する各モデルのロボット本体との関関係を設定します。 (モデルの位置が、ベース部か、Jn 軸か、またはフランジ部かを設定) 注）モデルの形状は球、および円筒です。 パラメータ：CAVKDA1 ～ 8	設定したハンド番号は、LoadSet (ロードセット) 命令のハンド条件番号と連動します。 注）モデルの形状は球、および円筒です。 パラメータ：CAVKDH1 ～ 8	設定したワーク番号は、LoadSet (ロードセット) のワーク条件番号と連動します。 注）モデルの形状は球、および円筒です。 パラメータ：CAVKDW1 ～ 8
		LoadSet 命令で干渉チェックの対象とするハンドとワークのモデルをダイナミックに変更できます。	

モデルの設定項目	モデルの種類		
	ロボット本体のモデル[注 1)]	ハンドのモデル	ワークのモデル
中心位置	ロボット本体据付面、または各軸の回転中心を基準とし、そこからの距離でモデルの中心位置指定します。 パラメータ：CAVPSA1 ～ 8	メカニカルインタフェース 座標系の原点（J3 軸の先端）からの距離でモデルの中心位置指定します。	
		パラメータ：CAVPSH1 ～ 8	パラメータ：CAVPSW1 ～ 8
モデルの大きさ	各モデルの大きさを半径で設定します。		
	パラメータ：CAVSZA1 ～ 8	パラメータ：CAVSZH1 ～ 8	パラメータ：CAVSZW1 ～ 8
各モデル毎の有効 / 無効	各モデルの有効 / 無効、および T/B 有効時に一時的に無効にするかどうかを設定します。 注）ハンドのモデル、ワークのモデルは教示操作時には無効にする必要があります。		
	パラメータ：CAVSCA1 ～ 8	パラメータ：CAVSCH1 ～ 8	パラメータ：CAVSCW1 ～ 8

注 1）工場出荷時に機種毎の初期値は設定してあります。

以下に各パラメータの説明を記載します。

(1) モデル登録用パラメータ

以下に前記表 5-31 でまとめた各パラメータの詳細を示します。
ロボット本体、ハンド、ワークにつきそれぞれ最大 8 個まで登録できる各々のモデルは、それぞれ各パラメータ名の下 1 桁の数字で区別します。

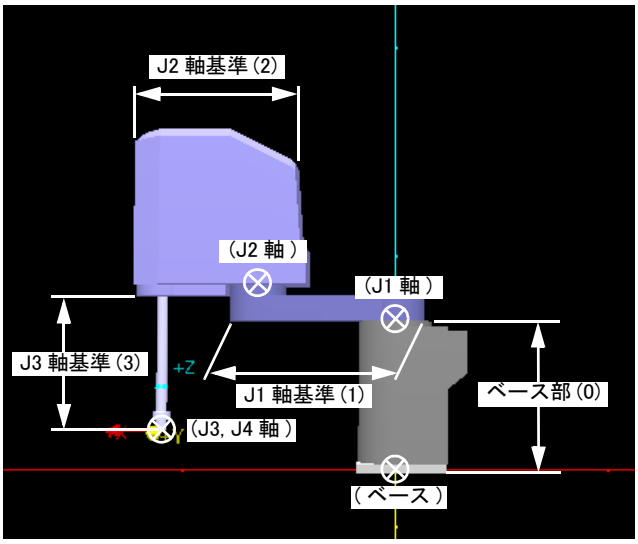
1) ロボット本体のモデル

①モデル登録部と形状：CAVKDA1 ～ 8

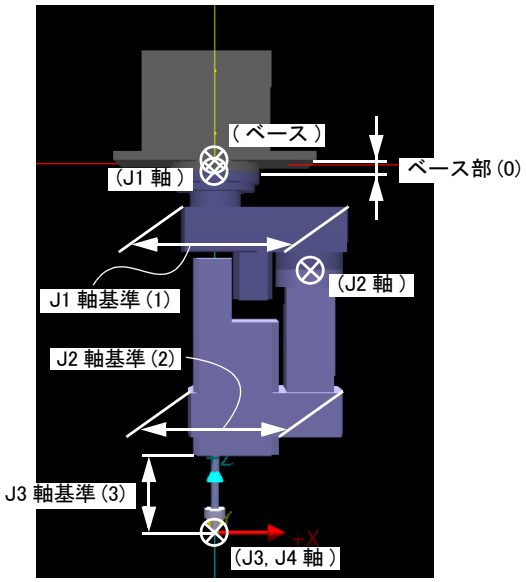
表 5-32：モデル設定パラメータ（ロボット本体：CAVKDA1 ～ 8）

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値	
モデル登録部と形状 （ロボット本体）	CAVKDA1 ～ 8	整数 2	モデルの登録部(基準軸)と形状を設定します。モデルは最大で8個まで登録できます。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:登録部(基準軸) 0 :ベース部 1～6 :Jn軸 第2要素:形状 0 :球 1 :円筒	RH-3/6/12/20FHシリーズ: CAVKDA1=0, 1 CAVKDA2=0, 0 CAVKDA3=1, 0 CAVKDA4=2, 0 CAVKDA5=2, 0 CAVKDA6=2, 0 CAVKDA7=2, 0 CAVKDA8=4, 1 注)RH-3FH35xx、RH-6FH35xx、RH-12FH55xxはCAVKDA3=0, 0です。 RH-3FHRシリーズ: CAVKDA1=0, 0 CAVKDA2=1, 1 CAVKDA3=1, 0 CAVKDA4=2, 1 CAVKDA5=2, 1 CAVKDA6=3, 1 CAVKDA7=0, 0 CAVKDA8=0, 0	RV-Fシリーズ: CAVKDA1=0, 0 CAVKDA2=0, 1 CAVKDA3=2, 1 CAVKDA4=4, 1 CAVKDA5=5, 0 CAVKDA6=0, 0 CAVKDA7=0, 0 CAVKDA8=0, 0 注)RV-2Fは CAVKDA2=0, 0 CAVKDA4=3, 0 CAVKDA5=4, 1 CAVKDA6=5, 0

<RH-3FH/6FH/12FH/20FH シリーズ>



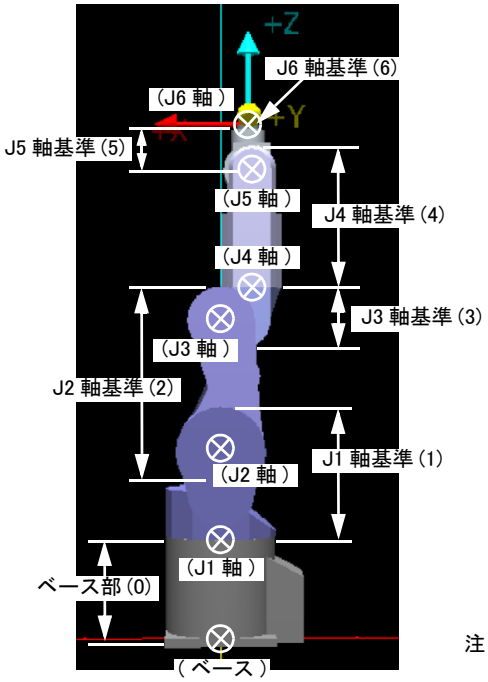
<RH-3FHR シリーズ>



注) カッコ内数字はパラメータ : CAVKDA1 ～ 8(モデル登録部と形状) の第 1 要素 (登録部 (基準軸)) 設定値

図 5-19 : RH-F シリーズモデル登録部 (補足)

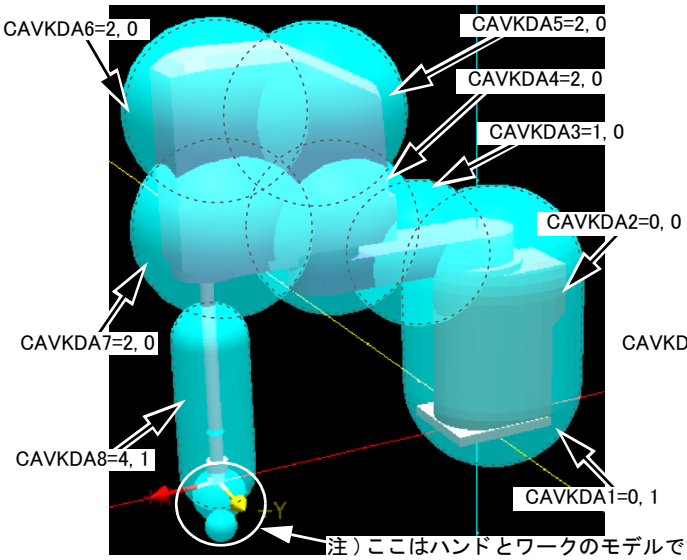
<RV-F シリーズ>



注) カッコ内数字はパラメータ : CAVKDA1 ～ 8(モデル登録部と形状) の第 1 要素 (登録部 (基準軸)) 設定値

図 5-20 : RV-F シリーズモデル登録部 (補足)

〈RH-F シリーズ〉



〈RV-F シリーズ〉

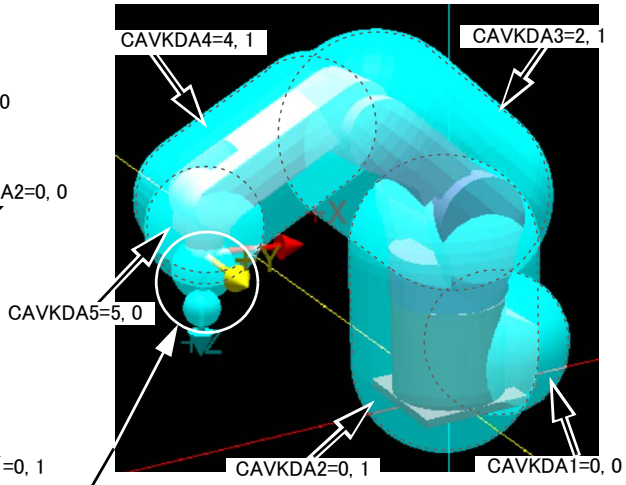
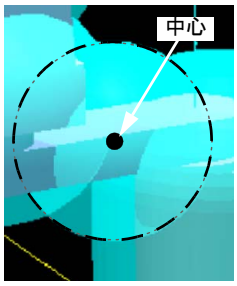
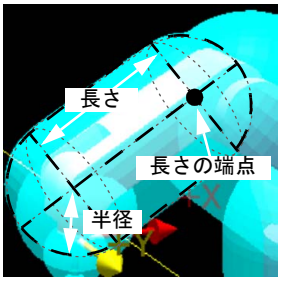


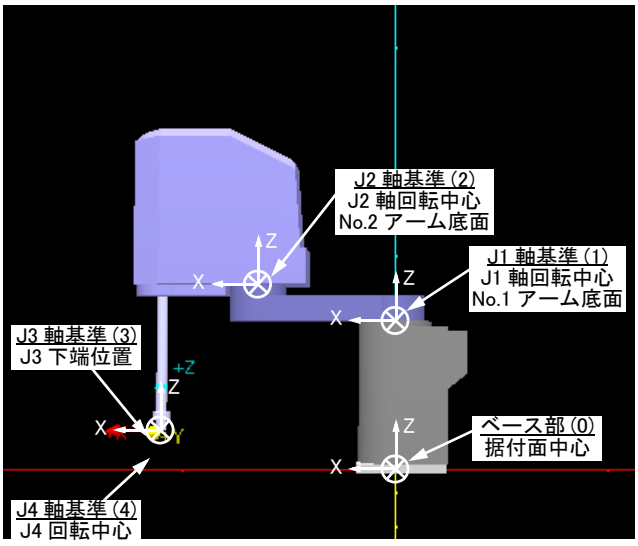
図 5-21：登録部とモデル形状の設定例

②モデルの位置：CAVPSA1 ～ 8

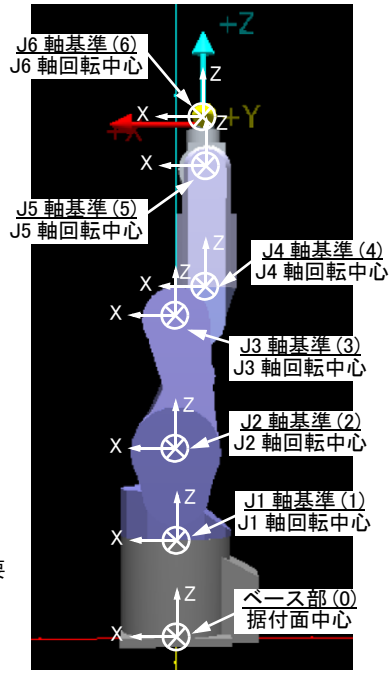
表 5-33：モデル設定パラメータ（ロボット本体：CAVPSA1 ～ 8）

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
モデルの位置 （ロボット本体）	CAVPSA1 ～ 8	実数 6	登録する各モデルの位置を、それぞれの基準軸位置からの距離と回転角で設定します。（各モデルはパラメータ名の下一桁の番号(1～8)に対応） 第1要素：X軸方向の距離(mm) 第2要素：Y軸方向の距離(mm) 第3要素：Z軸方向の距離(mm) 第4要素：X軸周りの回転角(deg) 第5要素：Y軸周りの回転角(deg) 第6要素：Z軸周りの回転角(deg) 注）回転角はZ軸→Y軸→X軸の順に反映されます。形状が球の場合、回転角の設定は不要です。 ＜設定するモデルの場所＞  球  円筒	

＜RH-F シリーズ＞



＜RV-F シリーズ＞



注）・括弧内の数字はパラメータ：CAVKDA1 ～ 8（モデル登録部と形状）の第 1 要素（登録部（基準軸））設定値
・上段：基準軸
下段：基準軸の場所

図 5-22：各モデルの基準軸位置 XYZ 方向

［補足］：各基準軸の XYZ 方向

RH-F シリーズ

J1,J2,J4 ＝ 0 度、J3 ＝下端の姿勢を想定したとき、各基準軸の XYZ 方向はベース座標系と一致します。

RV-F シリーズ

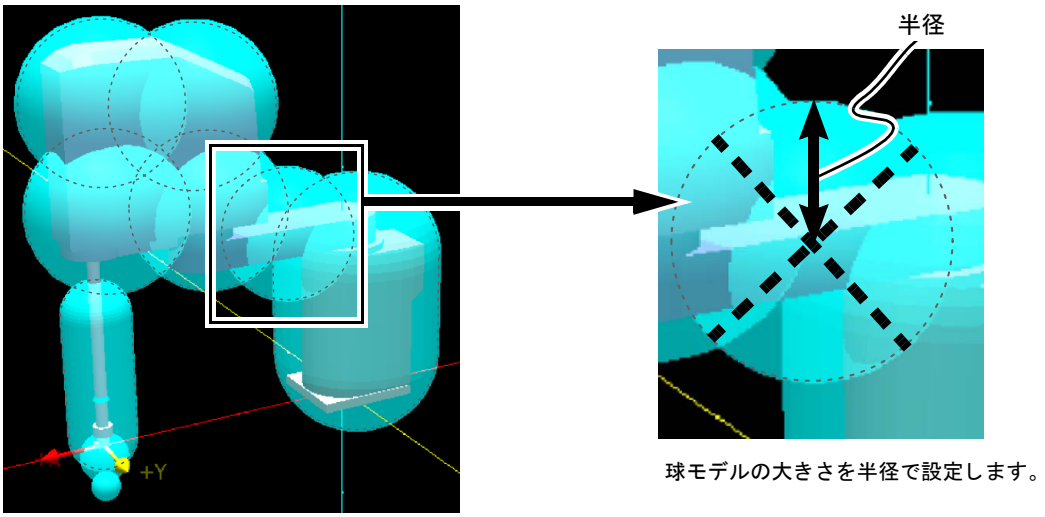
全軸が 0 度の姿勢を想定したとき、各基準軸の XYZ 方向はベース座標系と一致します。

③モデルの大きさ：CAVSZA1 ～ 8

表 5-34：モデル設定パラメータ（ロボット本体：CAVSZA1 ～ 8）

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
モデルの大きさ （ロボット本体）	CAVSZA1 ～ 8	実数 4	各モデルの大きさを設定します。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:半径(mm) 第2要素:長さ(mm) 第3要素:0に固定 第4要素:0に固定 注)形状が球の場合、長さの設定は不要です。	

〈RH-F シリーズ〉



〈RV-F シリーズ〉

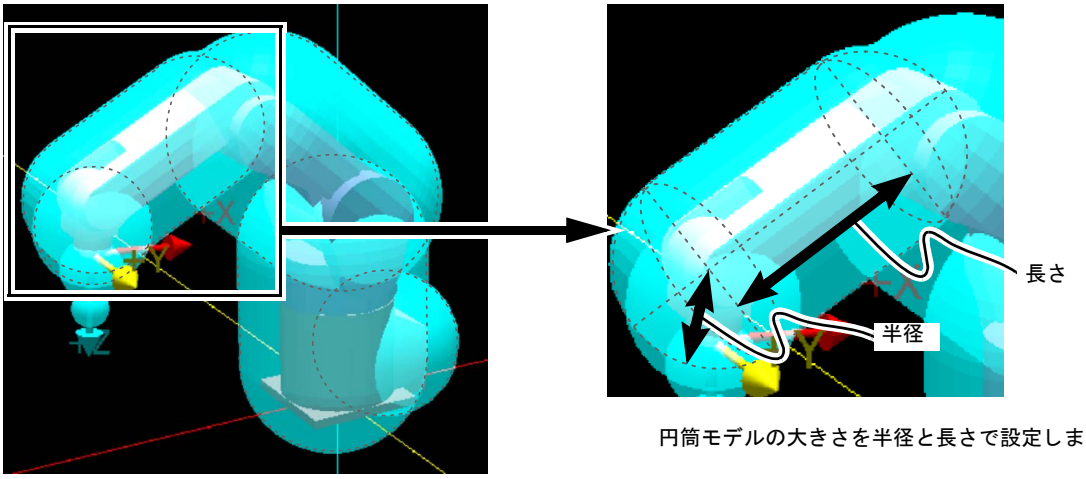


図 5-23：モデルの大きさ（補足）

④モデルの有効 / 無効 : CAVSCA1 ～ 8

表 5-35 : モデル設定パラメータ (ロボット本体 : CAVSCA1 ～ 8)

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値	
モデルの有効 / 無効 (ロボット本体)	CAVSCA1 ～ 8	整数 3	各モデル毎に干渉チェックの対象とするか否か(有効/無効)を設定します。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:有効/無効の設定(0:無効、1:有効) 第2要素:ジョグ操作における干渉回避機能の一時的解除操作で無効にするか否かの設定 (0:無効にする、1:有効状態を保つ) 注)教示時におこなうジョグ操作で、干渉せざるを得ないハンドやワークのモデルに"0:無効にする"を設定すると便利です。 第3要素:0に固定	RH-3/6/12/02FH シリーズ: CAVSCA1=1, 0, 0 CAVSCA2=0, 0, 0 CAVSCA3=1, 0, 0 CAVSCA4=1, 0, 0 CAVSCA5=1, 0, 0 CAVSCA6=1, 0, 0 CAVSCA7=1, 0, 0 CAVSCA8=1, 0, 0 RH-3FHRシリーズ: CAVSCA1=1, 0, 0 CAVSCA2=1, 0, 0 CAVSCA3=1, 0, 0 CAVSCA4=1, 0, 0 CAVSCA5=1, 0, 0 CAVSCA6=1, 0, 0 CAVSCA7=0, 0, 0 CAVSCA8=0, 0, 0	RV-Fシリーズ: CAVSCA1=1, 0, 0 CAVSCA2=1, 0, 0 CAVSCA3=1, 0, 0 CAVSCA4=1, 0, 0 CAVSCA5=1, 0, 0 CAVSCA6=0, 0, 0 CAVSCA7=0, 0, 0 CAVSCA8=0, 0, 0

2) ハンドのモデル

表 5-36 にハンドとして登録するモデルの設定パラメータを示します。

表 5-36 : モデル設定パラメータ (ハンド)

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ハンド番号と形状 (ハンド)	CAVKDH1 ～ 8	整数 2	ハンドとして登録する各モデルのハンド番号と形状を設定します。モデルは最大で8個まで登録できます。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:ハンド番号 LoadSet (ロードセット)命令でのモデル変更時のハンド条件番号に対応 0 :初期値として設定するモデル 1～8 :LoadSet (ロードセット)命令で指定するハンド条件番号 第2要素:形状 0 :球 1 :円筒	全てのパラメータ(CAVKDH1～8)を"0, 0"に設定。
モデルの中心位置 (ハンド)	CAVPSH1 ～ 8	実数 6	各モデル毎に、メカニカルインタフェース 座標系から見たモデルの位置と姿勢を指定します。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:X軸方向の距離(mm) 第2要素:Y軸方向の距離(mm) 第3要素:Z軸方向の距離(mm) 第4要素:X軸周りの回転角(deg) 第5要素:Y軸周りの回転角(deg) 第6要素:Z軸周りの回転角(deg) 注)回転角はZ軸→Y軸→X軸の順に反映されます。形状が球の場合、回転角の設定は不要です。	全てのパラメータ(CAVPSH1～8)を"0, 0, 0, 0, 0, 0,"に設定。
モデルの大きさ (ハンド)	CAVSZH1 ～ 8	実数 4	各モデルの大きさを設定します。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:半径(mm) 第2要素:長さ(mm) 第3要素:0に固定 第4要素:0に固定 注)形状が球の場合、長さの設定は不要です。	全てのパラメータ(CAVSZH1～8)を"0, 0, 0, 0"に設定。

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
モデルの有効 / 無効 (ハンド)	CAVSCH1 ～ 8	整数 3	各モデル毎に干渉チェックの対象とするか否か(有効/無効)を設定します。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:有効/無効の設定(0:無効、1:有効) 第2要素:ジョグ操作における干渉回避機能の一時的解除操作で無効にするか否かの設定(0:無効にする、1:有効状態を保つ) 注)教示時におこなうジョグ操作で、干渉せざるを得ないハンドやワークのモデルに"0:無効にする"を設定すると便利です。 第3要素:0に固定	全てのパラメータ(CAVSCH1～8)を"0, 0, 0"に設定。

◇◆◇プログラム実行時のチェック対象モデルの変更 (ハンド、ワーク) ◇◆◇

プログラム実行時には、実際に使用しているハンドや把持しているワークに合わせて干渉チェックをおこなうために、LoadSet 命令で干渉チェックの対象とするハンドとワークのモデルを変更できます。LoadSet 命令では予めパラメータで設定してあるハンド番号、ワーク番号を指定します。

3) ワークのモデル

表 5-37 にワークとして登録するモデルの設定パラメータを示します。

表 5-37 : モデル設定パラメータ (ワーク)

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ワーク番号と形状 (ワーク)	CAVKDW1 ～ 8	整数 2	ワークとして登録する各モデルのワーク番号と形状を設定します。モデルは最大で8個まで登録できます。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:ワーク番号 LoadSet (ロードセット)命令でのモデル変更時のワーク条件番号に対応 0 :初期値として設定するモデル 1～8 :LoadSet (ロードセット)命令で指定するワーク条件番号 第2要素:形状 0 :球 1 :円筒	全てのパラメータ(CAVKDW1～8)を"0, 0"に設定。
モデルの位置 (ワーク)	CAVPSW1 ～ 8	実数 6	各モデル毎に、メカニカルインタフェース 座標系から見たモデルの位置と姿勢を指定します。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:X軸方向の距離(mm) 第2要素:Y軸方向の距離(mm) 第3要素:Z軸方向の距離(mm) 第4要素:X軸周りの回転角(deg) 第5要素:Y軸周りの回転角(deg) 第6要素:Z軸周りの回転角(deg) 注)回転角はZ軸→Y軸→X軸の順に反映されます。形状が球の場合、回転角の設定は不要です。	全てのパラメータ(CAVPSW1～8)を"0, 0, 0, 0, 0, 0,"に設定。
モデルの大きさ (ワーク)	CAVSZW1 ～ 8	実数 4	各モデルの大きさを設定します。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:半径(mm) 第2要素:長さ(mm) 第3要素:0に固定 第4要素:0に固定 注)形状が球の場合、長さの設定は不要です。	全てのパラメータ(CAVSZW1～8)を"0, 0, 0, 0"に設定。
モデルの有効 / 無効 (ワーク)	CAVSCW1 ～ 8	整数 3	各モデル毎に干渉チェックの対象とするか否か(有効/無効)を設定します。(各モデルはパラメータ名の下一桁の番号(1～8)に対応) 第1要素:有効/無効の設定(0:無効、1:有効) 第2要素:ジョグ操作における干渉回避機能の一時的解除操作で無効にするか否かの設定(0:無効にする、1:有効状態を保つ) 注)教示時におこなうジョグ操作で干渉せざるを得ない、ハンドやワークのモデルに"0:無効にする"を設定すると便利です。 第3要素:0に固定	全てのパラメータ(CAVSCW1～8)を"0, 0, 0"に設定。

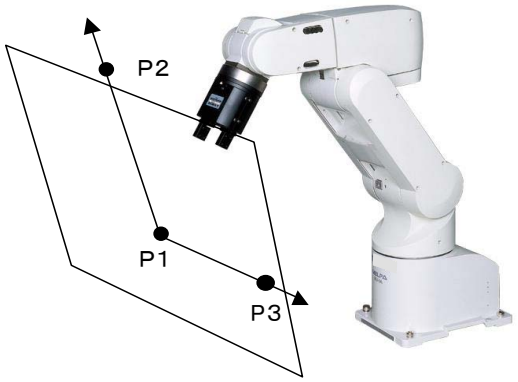
◇◆◇プログラム実行時のチェック対象モデルの変更（ハンド、ワーク）◇◆◇
プログラム実行時には、実際に使用しているハンドや把持しているワークに合わせて干渉チェックをおこなうために、LoadSet（ロードセット）命令で干渉チェックの対象とするハンドとワークのモデルを変更できます。
LoadSet（ロードセット）命令では予めモデルを定義するパラメータで設定してあるハンド番号、ワーク番号を指定します。

◇◆◇ワークは把持時のみ干渉チェックを実行◇◆◇
パラメータ HNDHOLD* で設定された把持状態の時のみ、ワークモデルを干渉チェックの対象とします。（ハンド開閉に連動）

5.24.4 自由平面リミットの登録

干渉チェックに使用する自由平面リミットを登録します。必要な登録内容を表 5-38 にまとめます。干渉チェックできる自由平面リミットは 8 個まで登録できます。

⚠注意 登録したロボット CPU のみで干渉チェックが行われます。



左図のように 3 点 (P1，P2，P3) にて任意の平面を定義します。

注）図は垂直 6 軸ロボットの例です。水平多関節型ロボットでも同じです。

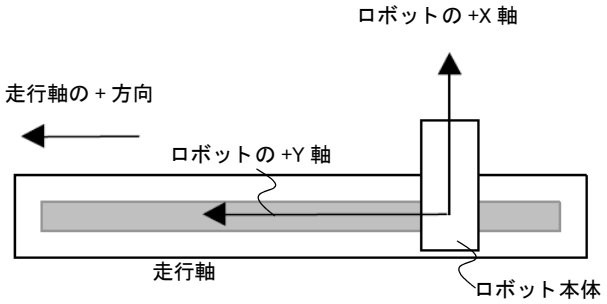
表 5-38：自由平面リミット登録内容

パラメータと値	説明
SFCnP(n=1 ～ 8)	平面を作成する為の 3 点を指定します。 P1 の座標値の　X1,Y1,Z1：平面の原点位置 P2 の座標値の　X2,Y2,Z2：平面の X 軸上の位置 P3 の座標値の　X3,Y3,Z3：平面の X-Y 平面上の +Y 方向の位置
CAVSCFn(n=1 ～ 8)	設定した自由平面リミットへの干渉チェック有効 / 無効を指定します。 0：無効（初期値） 1：有効

5.24.5 付加軸の対応

付加軸を使用しているロボットでは、付加軸の同期制御パラメータを設定することで付加軸の動きを考慮した干渉回避機能を使用することができます。(走行軸など直動軸のみ)
ご使用状態に合わせて表 5-39 を参照してパラメータを設定してください。

注) 後述の「5.24.7 ロボット間キャリブレーション (ロボット同士の干渉チェックの場合のみ)」で設定するロボット間の位置関係は、走行軸の座標値を "0" として設定してください。



この場合、パラメータ AXDIR = 0.0, 0.0, -90. 0 となります。
(ロボットの Z 軸周りに -90 度回転)

図 5-24 : 走行軸の使用例

表 5-39 : 付加軸の同期制御パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
干渉回避付加軸 (走行軸) 番号指定	CAVAXJNO	整数 1	干渉回避で考慮する付加軸(走行軸)の軸番号を設定。 7、8軸以外の設定は干渉チェック対象となりません。 設定値:0、7、または8	0
付加軸 同期方向	AXDIR	実数 3	走行軸の＋方向をX軸とした座標系からロボット座標系への変換 第1要素:X軸周りの回転角 第2要素:Y軸周りの回転角 第3要素:Z軸周りの回転角 注)工場出荷設定値では、ロボットのX軸は走行軸の＋方向と一致しています。	0.0, 0.0, 0.0

5.24.6 共有メモリ拡張機能の設定(ロボット同士の干渉チェックの場合のみ)

パラメータで共有メモリ拡張機能を設定します。

この設定により、ロボット CPU 間通信をおこなうための共有メモリを占有します。(図 5-25 参照)

(1) パラメータの設定

パラメータ：IQMEM のビット 4 を "1" にして、ロボット間協調制御機能を有効に設定します。
またパラメータ：QMLTCPUN にマルチ CPU システムで基本ベースユニットに装着している CPU ユニットの台数、およびパラメータ：QMLTCPUn の要素 1 に "2" を設定します。

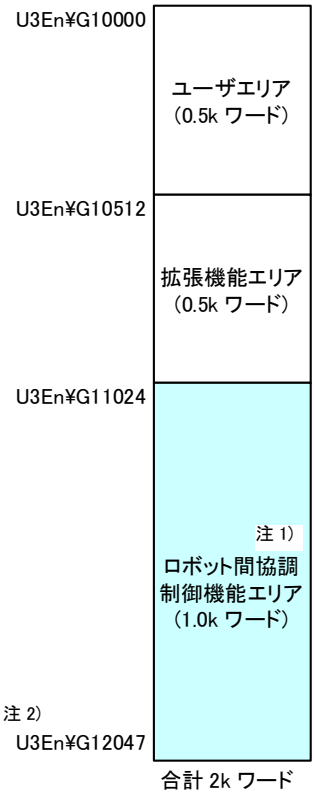
表 5-40：共有メモリ拡張機能選択パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
共有メモリ拡張機能選択	IQMEM	整数 1	共有メモリ拡張機能を選択します。 ビット毎に機能を割り当てています。1/0= 有効 / 無効 15 0 00000000 00000000 | ||...bit0: 拡張機能の使用 | |....bit1: シーケンサダイレクト実行機能 bit2-3、5-15 は未使用 | |.....bit4: ロボット間協調制御機能	00000000 00000000
マルチ CPU 台数設定	QMLTCPUN	整数 1	マルチCPUシステムで、基本ベースユニットに装着するCPUユニットの台数を設定します。	2
マルチ CPUn 号機高速通信エリア設定	QMLTCPUn n=1 ～ 4	整数 4	マルチCPUシステムで、1～4号機のマルチCPU間高速通信機能の各CPUユニット間で送受信をおこなう点数を設定します。 全CPUについてパラメータ設定値を合わせる必要があります。パラメータ設定値が合っていないとき、シーケンサCPUでエラーになりますので、各CPUのパラメータ設定値を合わせるようにしてください。 要素1:ユーザ自由エリアのサイズ(K点) 範囲:1～14(最大※) ※下表のようにマルチCPU台数によって最大値が異なります。 CPU 台数 / 設定範囲 2 / 0 ～ 14K 点 3 / 0 ～ 13K 点 4 / 0 ～ 12K 点 要素2:自動リフレッシュの点数(点) 範囲:0～14335 ロボットCPUは、自動リフレッシュに対応しておりませんので、自動リフレッシュの点数は、**常に0としてください。** 要素3:システムエリアのサイズ(K点) 範囲:1または2 要素4:マルチCPU同期立ち上げ(1:する、0:しない) ロボットCPUは、立ち上がりに時間がかかりますので、基本的に1(同期する)のままで、**設定を変えないでください。**	1,0,1,1

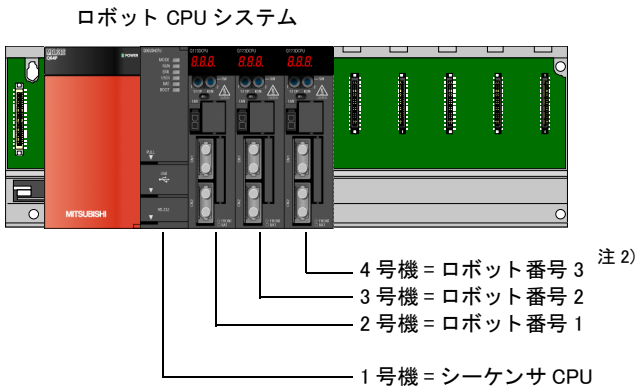
(2) 共有メモリマップ

パラメータ：IQMEM の設定状態に基づく共有メモリ上のロボット CPU 出力エリアのメモリマップを示します。

＜共有メモリマップ＞



＜スロットとロボット番号の関係＞



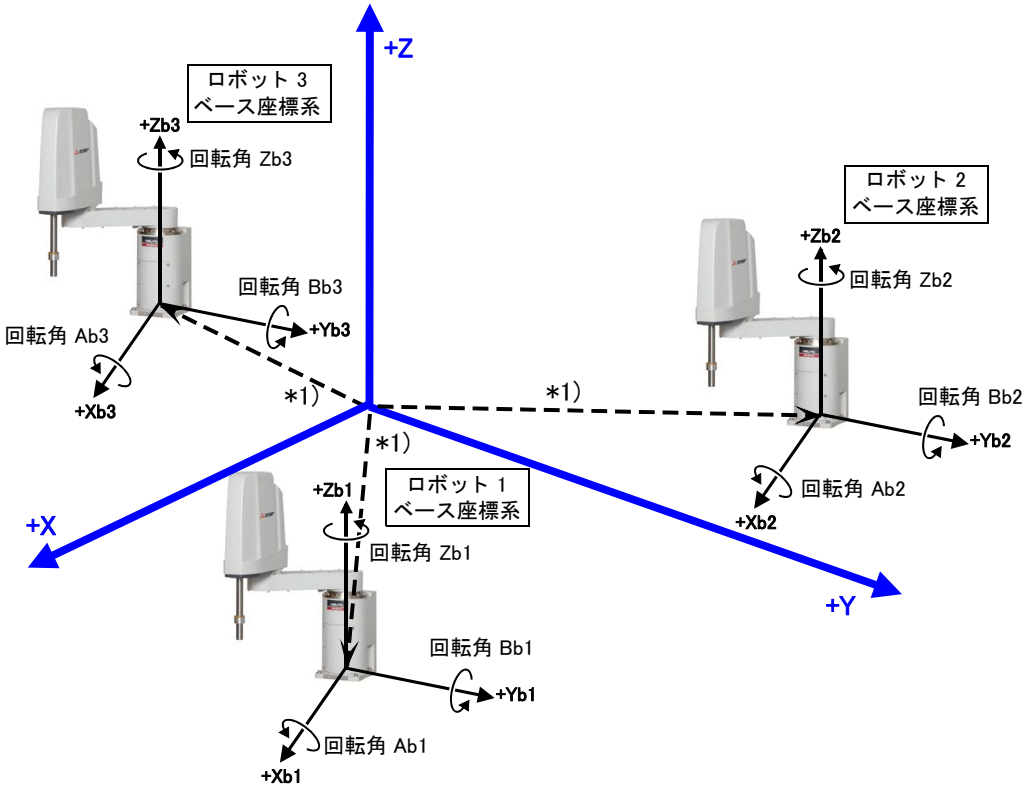
注 1) ロボット間協調制御機能のアドレスは、拡張機能の有効 / 無効状態にかかわらず固定です。

注 2) 共有メモリアドレスの表記にある U3En¥ の "n" は、ロボット番号に対応します。
ロボット番号は、ロボット CPU システム (iQPlatform) のスロット位置毎に固定です。

図 5-25：共有メモリマップ

5.24.7 ロボット間キャリブレーション(ロボット同士の干渉チェックの場合のみ)

干渉回避機能を有効とするロボット間の位置関係を設定します。
ロボット間で共通の座標系原点を、装置のレイアウト図などで一つ決定します。その座標系から見た各ロボットのベース座標系原点位置をパラメータ：RBCORD に設定します。



*1) 共通の座標系から見た各ロボットのベース座標系原点。
この座標値をそれぞれのロボットのパラメータ：RBCORD に設定。

図 5-26：ロボット間キャリブレーションイメージ図

(1) キャリブレーションの設定

ロボット間共通座標から見た各ロボットのベース座標系原点位置をパラメータ：RBCORD に X, Y, Z, A, B, C の座標値で設定します。

注）走行軸をご使用の場合は、走行軸の座標値を "0" とした時の位置関係を設定してください。

表 5-41：ロボット間キャリブレーション設定パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ロボット間共通座標	RBCORD	実数 6	ロボット間共通座標から見た自ロボットのベース座標系原点位置。 (X, Y, Z, A, B, Cの座標値で指定) 第1要素:X軸座標値(mm) 第2要素:Y軸座標値(mm) 第3要素:Z軸座標値(mm) 第4要素:A軸座標値(deg) (X軸周りの回転角) 第5要素:B軸座標値(deg) (Y軸周りの回転角) 第6要素:C軸座標値(deg) (Z軸周りの回転角) 注1)A, B, C軸座標値(回転角)は、Z軸周り→Y軸周り→X軸周りの順に回転させて得られる値を設定してください。	0.00, 0.00, 0.00, 0.00, 0.00, 0.00,

(2) キャリブレーションの設定結果の確認

それぞれのロボットのキャリブレーション設定が正しくおこなわれていることを確認します。

確認手順を以下に示します。

1) 装置のレイアウト図などで基準とする位置を各ロボット毎に一箇所決定します。(図 5-27 の a)。以下基準点)
2) 各ロボットのハンド先端を決定した基準点にジョグ操作で合わせます。
3) T/B などでロボット (システム) 状態変数 : P_CurrR(共通の座標系から見たロボットの現在位置) の値を確認します。
4) 上記 P_CurrR の値と、レイアウト図などから読み取った値を比較します。
比較対象は XYZ の値のみです。双方が合っていればキャリブレーションは正しくおこなわれています。異なっている場合はパラメータ : RBCORD の設定値を修正してください。
5) 上記操作を干渉回避機能を有効とするロボットすべてについて実施します。

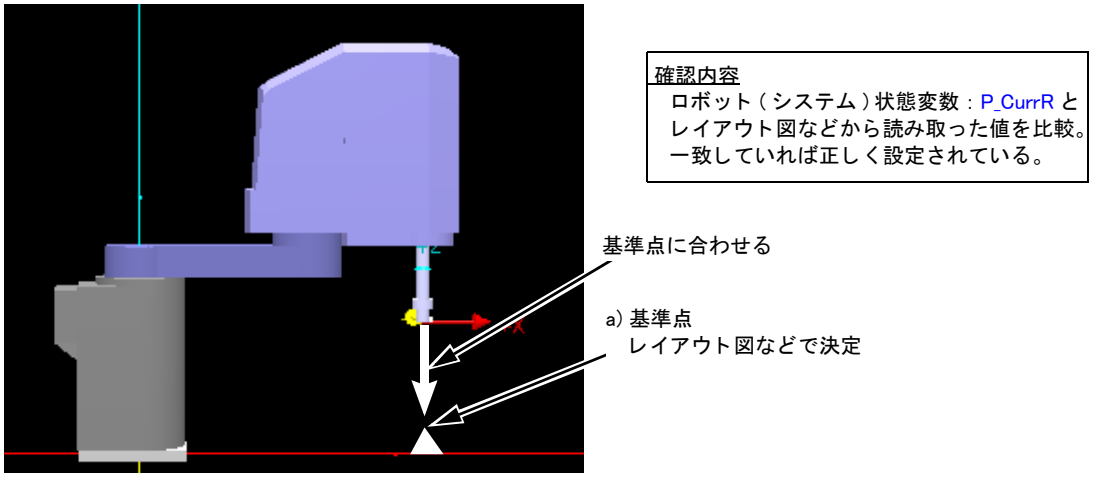


図 5-27 : ロボット間キャリブレーション設定の確認

5.24.8 干渉回避機能の有効/無効設定

干渉回避機能自体の使用可能 / 不可能と、プログラム実行時、およびジョグ操作時それぞれの有効 / 無効をパラメータ：CAV で設定することができます。
表 5-42 に設定するパラメータの詳細を示します。

表 5-42：干渉回避機能の有効 / 無効設定パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
干渉回避機能設定	CAV	整数 3	干渉回避機能自体の使用可能/不可能を設定します。 また、使用可能に設定した場合、プログラム実行時の有効/無効、およびジョグ操作時の有効(含む干渉時の動作)/無効を設定します。 第1要素:干渉回避機能使用の設定(0:使用不可能/1:使用可能) 第2要素:プログラム運転時の初期状態を指定(0:無効、1:有効) 第3要素:ジョグ操作時の有効/無効設定 (0:無効、1:有効・エラー発生、2:有効・ブザー音のみ)	0, 1, 1

5.24.9 干渉回避機能の使用

以下にジョグ操作時、およびプログラム実行時の干渉回避動作を説明します。パラメータ CAV の第 1 要素は "1"(使用可能) に設定しておきます。(498 ページの「5.24.8 干渉回避機能の有効 / 無効設定」参照)

(1) ジョグ操作時の干渉回避

パラメータ : CAV の第 3 要素を有効 ("1" または "2") に設定すると、ジョグ操作時に干渉回避機能を使用することができます。

干渉を検知するとロボットは減速停止し、同パラメータの設定値によって以下の動作に区別されます。

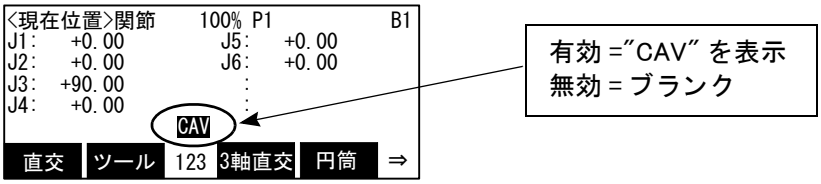
設定値 =1 : アラーム (L241n) が発生します。ジョグ操作を再開するためには一度アラームリセット操作が必要です。

設定値 =2 : ブザーが鳴ります。干渉しない方向へのジョグ動作は可能です。

干渉回避機能の一時的解除操作

モデルの有効 / 無効設定パラメータ : CAVSCA1 ～ 8(ロボット本体)、CAVSCH1 ～ 8(ハンド)、CAVSCW1 ～ 8(ワーク) の各第 2 要素を "0"(一時的に無効にする) に設定したモデルは、T/B の [CLEAR] キーを押すことによって一時的に干渉チェックを無効にすることができます。

[CLEAR] キーを押す毎に有効 / 無効が切り替わり、有効状態では T/B 画面中央下に "CAV" を表示します。



なお、[CLEAR] キーによる操作でも干渉回避から復帰できない場合 (サーボ OFF 中に人手でロボットを干渉領域に入れてしまった場合など) は、本書ページ 61 ページの「3.10 解除できないエラーの一時的エラーリセット操作」([RESET] キーを押したままジョグ操作) をおこなうことで一時的に干渉回避機能自体を解除してロボットを動作させることができます。

注) 但し、この操作では動作範囲リミットでロボットは停止しません。動作範囲外の方向へ動かさないようご注意願います。

(2) プログラム実行時の干渉回避

パラメータ : CAV の第 2 要素を有効 ("1") に設定し、表 5-43 に示す命令語、および表 5-44 に示す外部変数を利用することでプログラム実行時 (自動運転中) に干渉回避機能を使用することができます。

表 5-43 : 命令語 (干渉回避機能)

命令語	内容	解説ページ
CavChk	干渉回避機能の有効/無効設定と、干渉検知時のアラーム発生有無を設定します。 CavChk <ON/OFF>[, <ロボットCPU番号>[,NOErr]]	174 ページ

表 5-44 : ロボット (システム) 状態変数 (干渉回避機能)

変数名	内容	解説ページ
P_CordR	共通座標系から見た自ロボットのベース座標系原点位置 データの型 : 位置型	357 ページ
P_CurrR	共通座標系から見た自ロボットの現在位置 データの型 : 位置型	359 ページ
P_CavDir	干渉を検知したときのロボットの動作方向 データの型 : 位置型	355 ページ
M_CavSts	干渉状態。 0 : 干渉なし 1 ～ 3 : 干渉を検知した時の干渉相手のロボット番号 データの型 : 整数型	302 ページ

【補足】
干渉を検知するとロボットは減速停止し、CavChk On（キャブチェック オン）命令での "NOErr" の指定有無によって以下の動作に区別されます。
NOErr の指定なし：アラーム（L240n）が発生します。自動運転を再開するためには一度アラームリセット操作が必要です。
NOErr の指定あり：アラームは発生せず、プログラムで指定された割込み処理を実行します。
注）NOErr を指定する場合は、干渉を検知した時に処理する割込みプログラムが必ず必要です。
本命令を実行する前に割込み処理が定義されていなければ、本命令実行時にアラームが発生します。
割込み処理の概要：ロボット（システム）状態変数 M_CavSts の値が "0" でなくなったら干渉を検知しています。この状態で割込み処理をおこなうよう宣言します。
500 ページの「5.24.10 サンプルプログラム」にサンプルプログラムを示します。

⚠注意　原点設定が完了していないロボットは、現在の関節座標値を常に全軸 0 度として干渉チェックをおこないます。
このため本機能は正しく働きません。必ず原点設定を完了させてから本機能を実行してください。

5.24.10 サンプルプログラム
(1) 干渉回避機能の開始と終了（全ロボット）
＜プログラム例＞　注）ステップ番号は割愛しています。

```
:
'--- 初期状態（干渉回避機能有効）---; パラメータ CAV で干渉チェックが有効に設定されている場合。
:
MVS P1          ' 干渉回避機能有効での動作
:
'--- 全ロボットの干渉回避機能無効 ---
CavChk OFF      ' ロボット番号の省略で全ロボットを対象に無効
:
MVS P2          ' 干渉回避機能無効での動作
:
'--- 全ロボットの干渉回避機能有効 ---
CavChk ON       ' ロボット番号の省略で全ロボットを対象に有効
:
MVS P3          ' 干渉回避機能有効での動作
:
```

(2) 干渉回避機能の開始と終了（ロボットを指定）
＜プログラム例＞　注）ステップ番号は割愛しています。

```
:
'--- 2 号機との干渉回避機能を終了させる ---
CavChk OFF,2    ' ロボット番号 2 との干渉回避機能を終了
:
MVS P5          ' ロボット番号 2 との干渉回避機能無効での動作
:
'--- 2 号機との干渉回避機能を開始させる ---
CavChk ON,2     ' ロボット番号 2 との干渉回避機能を再開
:
MVS P6          ' ロボット番号 2 との干渉回避機能有効での動作
:
```

(3) ハンド、ワークのモデルを変更

＜プログラム例＞　注）ステップ番号は割愛しています。

```
   :
'--- ハンドのモデルを変更 ---
LoadSet 2,2       ' ハンド･ワークのモデル 2 を指定
   :
MVS P7            ' ハンド･ワークモデル 2 での干渉回避機能を実行
   :
'--- 干渉チェックのモデル変更（ワーク）---
LoadSet 1,2       ' ハンドモデル 1、ワークモデル 2 を指定
   :
MVS P8            ' ハンドモデル 1、ワークモデル 2 での干渉回避機能を実行
   :
'--- 干渉チェックのモデル変更（ワーク）---
LoadSet 0,0       ' ハンド･ワークモデルを初期値に戻す
   :
MVS P9            '
   :
```

(4) 干渉を検知したら待避動作を実行（割込み処理）

＜プログラム例＞　注）ステップ番号は割愛しています。

```
Def Act 1,M_CavSts<>0 GoTo *Home,S  ' 干渉を検知したときに実行する処理を割込で定義
Act 1=1
CavChk On,0,NOErr       ' エラー非発生モードで干渉回避機能を有効にする
Mov P1                  ' 干渉回避機能有効での動作
Mov P2                  ' 干渉回避機能有効での動作
Mov P3                  ' 干渉回避機能有効での動作
   :
   :
*Home                   ' 干渉を検知すると割り込むステップ
CavChk Off
M_CavSts=0              ' 干渉状態をクリア
MDist=Sqr(P_CavDir.X*P_CavDir.X+P_CavDir.Y*P_CavDir.Y+P_CavDir.Z*P_CavDir.Z)
                        ' 移動量の比率を求める
PESC=P_CavDir(1)*(-50)*(1/MDist)
                        ' 待避動作の移動量を作成（干渉位置から 50mm 戻る）
PDST=P_Fbc(1)+PESC      ' 待避位置を作成する
Mvs PDST                ' 待避動作
Mvs PHome               ' 待避位置へ移動
   :
```

5.25 シーケンサ入出力ユニット直接制御

F-Q シリーズのコントローラにおいて、ベースユニットにスロットインされた他号機が管理する入出力ユニット / 入出力混合ユニットの入出力信号の参照と、ロボット CPU が管理する出力ユニット / 入出力混合ユニットを指定し、直接信号出力※1) をおこなうことができます。

以下にその仕様と手順の概略を示します。

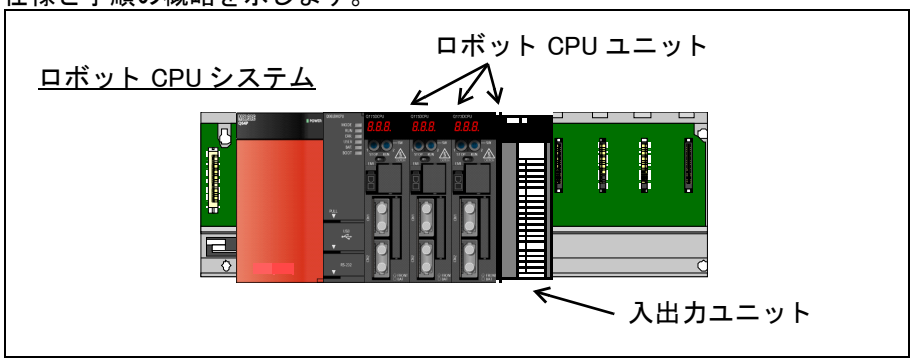


図 5-28 : シーケンサ入出力ユニット直接制御

(1) 仕様

本機能は、パラメータ QXYREAD で他号機が管理する入出力ユニット / 入出力混合ユニットの入出力信号参照の有効 / 無効の切り替えと、パラメータ QXYUNIT でロボット CPU が管理対象とする入出力ユニットを指定して使用します。表 5-45 に仕様を示します。

表 5-45 : 仕様

No.	項目	仕様
1	対象ソフトウェアバージョン	本機能は以下のソフトウェアバージョン以降で有効です。 本体 : R2b 注) R32TB/R33TB, R56TB/R57TB, RT ToolBox2 のソフトウェアバージョンの制限はありません。
2	対象スロット	基本ベースおよび増設ベース (7 段) の全てのスロットに対応。
3	対象ユニット	三菱 iQ Platform 対応シーケンサ MELSEC-Q シリーズが対応する全入出力ユニットに対応。
4	指定デバイス範囲	入力信号 : X0 ～ XFFF(4096 点) 出力信号 : Y0 ～ YFFF(4096 点)

(2) 操作手順の概要

1) 入出力信号を参照する場合

(1) パラメータの設定 … ロボットのパラメータ QXYREAD を設定します。(他号機が管理する入出力ユニットの信号取込有効 / 無効を設定)

↓

(2) シーケンサの " マルチ CPU" 設定 … " グループ外の入出力設定 " にチェックを入れ、グループ外の入力、出力状態とも取り込みを可能にします。

↓

(3) コントローラ電源の OFF → ON … コントローラの電源を一度入れなおします。

↓

(4) 以下、プログラムの状態変数 "M_XDev*() : 入力信号 (X) の参照 "/"M_YDev*() : 出力信号 (Y) の参照 " で入出力信号の参照が可能です。

※1) ロボットが管理 CPU となった信号をシーケンサ CPU(もしくはシーケンサ CPU に接続した GX Developer/GX Works) で参照するには、" グループ外の入力・出力状態を取り込む " の設定を有効にする必要があります。詳細は「(4) シーケンサの " マルチ CPU 設定 "」を参照願います。

2) 出力信号を出力する場合

(1) パラメータの設定 … ロボットのパラメータ QXYUNIT を設定します。
(ロボット CPU が管理対象とする入出力ユニットを設定)

↓

(2) コントローラ電源の OFF → ON … コントローラの電源を一度入れなおします。

↓

(3) 以下、プログラムの状態変数 "M_YDev*() : 出力信号 (Y) の書込み " で出力信号の直接出力が可能です。

(3) パラメータの説明

表 5-46 に関連するパラメータの概要を説明します。詳細は 419 ページの「5.2 信号パラメータ」の各説明頁を参照願います。

表 5-46 : 関連するパラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	解説ページ
シーケンサ入出力信号の取込設定	QXYREAD	整数 2	他号機 CPU が管理する入出力ユニット (入出力混合ユニット含む) の入出力信号参照機能の有効 / 無効を選択します。 入力信号 (X) と出力信号 (Y) を分けて設定できます。	421 ページ
シーケンサ入出力ユニット設定	QXYUNITn (n : 1 ～ 8)	整数 7	ロボット CPU が管理 CPU とする入出力ユニット / 入出力混合ユニットを設定します。 ユニットタイプ、先頭入出力番号、ベース番号、スロット番号、入出力点数、エラー時出力モード、応答時間を指定します。	421 ページ

(4) シーケンサの " マルチ CPU 設定 "

シーケンサ側の " マルチ CPU 設定 " の " グループ外の入出力設定 " にある " グループ外の入力状態を取り込む " と、" グループ外の出力状態を取り込む " の両方にチェックを入れて、グループ外の入力状態、出力状態とも取り込みを可能にします。

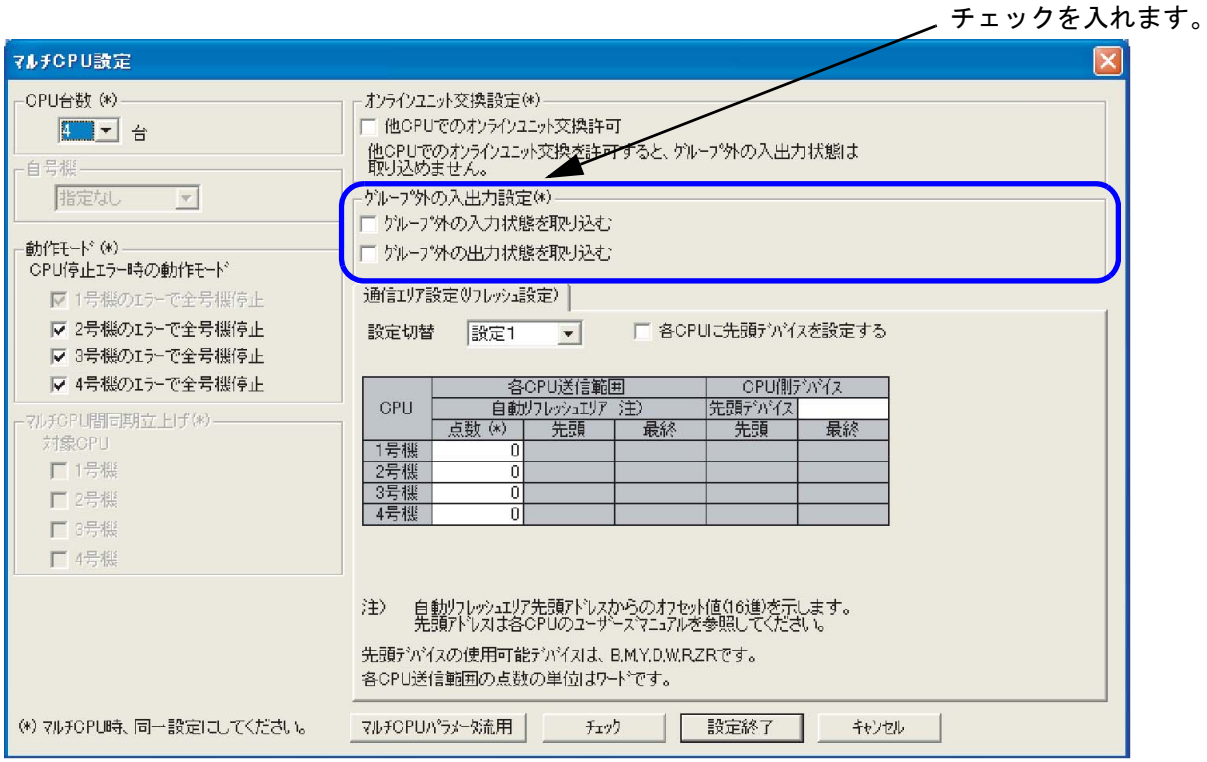


図 5-29 : GX Developer の PC パラメータのマルチ CPU 設定画面

【補足】

ロボットが管理 CPU となった信号をシーケンサ CPU(もしくはシーケンサ CPU に接続した GX Developer/GX Works) で参照するには " グループ外の入力・出力状態を取り込む " の設定を有効にする必要があります。

(機能の詳細内容はシーケンサの取扱説明書を参照願います)

(5) ロボット状態変数の説明

表 5-47 に関連するロボット状態変数の概要を説明します。詳細は 288 ページの「4.14.2 各ロボット状態変数 (システム状態変数) の説明」の各説明頁を参照願います。

表 5-47 : 関連するロボット状態変数

変数名	内容	解説ページ
M_XDev	シーケンサ入力信号 (X) をビット単位で読出します。 例)1 M1%=M_XDev(1) 'M1 にシーケンサ入力信号 1 番の値 (1 か 0) を代入	349 ページ
M_XDevB	シーケンサ入力信号 (X) をバイト単位で読出します。 例)1 M2%=M_XDevB(&H10) 'M2 にシーケンサ入力信号 10 番 (16 進数) から 8 ビット分の値を代入	349 ページ
M_XDevW	シーケンサ入力信号 (X) をワード単位で読出します。 例)1 M4%=M_XDevW(&H20) 'M4 にシーケンサ入力信号 20 番 (16 進数) から 16 ビット分の値を代入	349 ページ
M_XDevD	シーケンサ入力信号 (X) をダブルワード単位で読出します。 例)1 M5&=M_XDevD(&H100) 'M5 にシーケンサ入力信号 100 番 (16 進数) から 32 ビット分の値を代入	349 ページ
M_YDev	シーケンサ出力信号 (Y) をビット単位で読出し / 書込みます。 例)1 M_YDev(2)=1 Dly 0.5 ' シーケンサ出力信号 2 番を 0.5 秒間 ON します (パルス出力)	351 ページ

変数名	内容	解説ページ
M_YDevB	シーケンサ出力信号 (Y) をバイト単位で読出し / 書込みます。 例)1 M_YDevB(&H10)=&HFF　　' シーケンサ出力信号 10 番 (16 進数) から 8 ビット分を ON します	351 ページ
M_YDevW	シーケンサ出力信号 (Y) をワード単位で読出し / 書込みます。 例)1 M_YDevW(&H20)=&HFFFF　　' シーケンサ出力信号 20 番 (16 進数) から 16 ビット分を ON します	351 ページ
M_YDevD	シーケンサ出力信号 (Y) をダブルワード単位で読出し / 書込みます。 例)1 M_YDevD(&H100)=P1.X * 1000　　' シーケンサ出力信号 100 番 (16 進数) から 32 ビット分に位置変数 P1 の X 座標値を 1000 倍して格納します	351 ページ

5.26 ロボットCPU間直接通信

F-Q シリーズのコントローラにおいて、複数のロボット CPU 間で信号のやりとりを直接おこなう機能です。
シーケンサのラダープログラムを介さずに、複数のロボット間でインターロックなどのインタフェースをより高速に取ることができます。また、モーション CPU などロボット CPU 以外の共有メモリ情報の取込みも可能です。

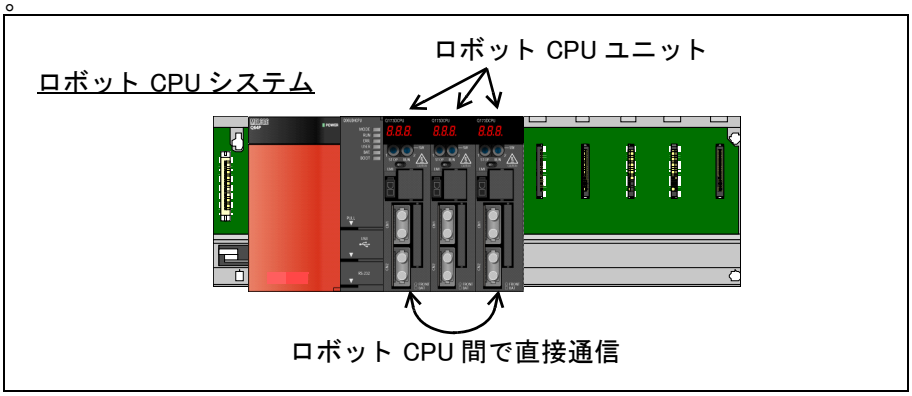


図 5-30：ロボット CPU 間直接通信

(1) 仕様

本機能は、表 5-48 に示す対象ソフトウェアバージョン以降に組み込まれており、パラメータを設定することなくご使用いただけます。
表 5-48 に仕様を示します。

表 5-48：仕様

No	項目	仕様
1	対象ソフトウェアバージョン	本機能は以下のソフトウェアバージョン以降で有効です。 本体：R2b 注）R32TB/R33TB，R56TB/R57TB，RT ToolBox2 のソフトウェアバージョンの制限はありません。
2	指定デバイス範囲	マルチ CPU 間共有デバイス（U3En¥G10000 ～ G24335）で指定できる範囲。
3	専用入出力信号の割付	専用信号によるロボットの制御は、シーケンサ（1 号機）で実施します。 2 号機以降の共有メモリデバイスへ専用入出力パラメータを割付けることはできません。 専用入出力信号を共有メモリデバイスへ割り付けてご使用になる場合は、必ずシーケンサ（1 号機）の共有メモリを割り付けてご使用ください。

(2) 利用方法

プログラムの状態変数 "M_UDevW：ワード単位（16 ビット分）で書き込み / 参照 "、"M_UDevD：ダブルワード単位（32 ビット分）で書き込み / 参照 " を使用することで、マルチ CPU 間共有デバイス（U3En¥G □）の読出し / 書込みが可能となります。

(3) ロボット状態変数の説明

表 5-49 に関連するロボット状態変数の概要を説明します。詳細は 288 ページの「4.14.2 各ロボット状態変数（システム状態変数）の説明」の各説明頁を参照願います。

表 5-49：関連するロボット状態変数

変数名	内容	解説ページ
M_UDevW	マルチ CPU 間共有デバイス（U3En¥G □）をワード単位で読出し / 書込みます。 例）1 M_UDevW(&H3E1, 10010)=&HFFFF　'2 号機 CPU の共有メモリアドレス 10010 に &HFFFF(16 進数）を書き込みます。	342 ページ
M_UDevD	マルチ CPU 間共有デバイス（U3En¥G □）をダブルワード単位で読出し / 書込みます。 例）1 M_UDevD(&H3E1, 10011)=P1.X * 1000　'2 号機 CPU の共有メモリアドレス 10011/10012 の 2 ワードに位置変数 P1 の X 座標値を 1000 倍した値を書き込みます。	342 ページ

5.27 2重系エラーの挙動選択パラメータについて

表 5-50 の二重系エラーに関して、パラメータ：CATEGORY を設定することにより、エラー発生時における [RESET] キー入力時の挙動を選択することができます。

表 5-50：対象エラー一覧

エラー番号	内容
H0039	ドアスイッチ開放信号配線異常
H0046	イネーブリングデバイス配線異常
H0051	外部非常停止配線異常

[RESET] キー入力時の挙動は、表 5-51 のいずれかを選択します。

表 5-51：パラメータ設定値に対する挙動

パラメータ設定値	内容
3(エラーリセット可)	エラーリセットが可能となる。(初期値)
4(エラーリセット不可)	エラーリセットが不可能となる。

本設定内容を反映するために、必ずコントローラの電源を再投入してください。

[注意]
本設定は、関連する信号線や配線の異常が解消された後の挙動を選択するものです。
信号線および配線の異常が残存している場合、本パラメータの設定にかかわらず、上記のエラーは解除されません。

6 外部入出力の機能

6.1 種類

(1) 専用入出力.....ロボットのプログラム実行、停止などの遠隔操作、実行中情報、サーボ電源状態などの状態表示をする入出力信号です。
各入出力信号に機能を割り付けて使用します。割付方法は各専用のパラメータに使用信号番号を設定して使用するもの(509 ページの「6.2 シーケンサリンク入出力機能」参照。)と、非常停止入力(544 ページの「6.7 非常停止入力」参照。)があります。あらかじめ使用頻度の高い信号が割り付けてあります。追加、変更も可能です。

(2) 汎用入出力.....ロボットのプログラムにてシーケンサなどとのやりとりに使用します。周辺装置の位置決め信号を取り込んだり、ロボットの位置を確認したしりする場合に使用します。

(3) ハンド入出力.....ハンド用の制御信号です。ハンドの開閉指示やハンドについているセンサ情報の取り込みに使用します。ユーザプログラムにて制御可能です。ロボットのハンド先端付近まで配線がされています。

表 6-1：入出力信号全体マップ

	入出力信号番号	使用方法
ハンド入出力	900 ～ 907	M_In, M_Inb, M_Inw, M_Out, M_Outb, M_Outw 変数で参照 / 代入 HOpen, HClose 命令でも可 例)If M_In(900)=1 Then M_Out(900)=1 HOpen 1, HClose 1
シーケンサリンク入出力	10000 ～ 18191	M_In, M_Inb, M_Inw, M_Out, M_Outb, M_Outw 変数で参照 / 代入 例)If M_In(10080)=1 Then M_Out(10080)=1 注) 専用出力が割り付けてある信号には、M_Out, M_Outb, M_Outw 変数を用いて出力することはできません。

6.2 シーケンサリンク入出力機能

本機能は、CR7xx-Q シリーズコントローラでのみ有効です。QnUD(H)CPU(以降シーケンサ CPU と呼びます) および、Q172DRCPU(以降ロボット CPU と呼びます) は、CPU 間の共有メモリを使用して、シーケンスラダープログラムにより交信します。交信には、共有メモリの「マルチ CPU 間高速通信エリア※1)」を使用します。ロボット CPU は、入力信号、出力信号ともに 10000 ～ 18191 までの信号番号を使います。

6.2.1 パラメータ設定

シーケンサリンク機能を使用するためには、シーケンサ CPU および、ロボット CPU、それぞれについて、マルチ CPU 関連のパラメータを設定する必要があります。
ロボット CPU は、RT ToolBox またはティーチングボックス (R33TB、R57TB)、シーケンサ CPU は、GX Developer を用いてパラメータ設定します。詳細については、各設定ツールの取扱説明書を参照してください。

(1) シーケンサ CPU のパラメータ設定

GX-Developer を用いて、マルチ CPU のパラメータ設定をおこないます。下記は、シーケンサ CPU1 台、ロボット CPU1 台のシステム時の設定例です。

1) CPU 台数
マルチ CPU システムで基本ベースユニットに装着する CPU ユニットの台数を設定します。

2) マルチ CPU 間同期立ち上げ
ロボット CPU は電源を入れてからシステムが立ち上がるまでに 10 数秒かかります。そのためマルチ CPU システムで同時に立上げる設定 (チェックを入れた状態) にしてください。

3) マルチ CPU 間高速通信エリア設定
点数を K ワード単位で設定します。ロボット CPU については、1K ワード未満しか使いませんので、1K ワードの設定にしてください。

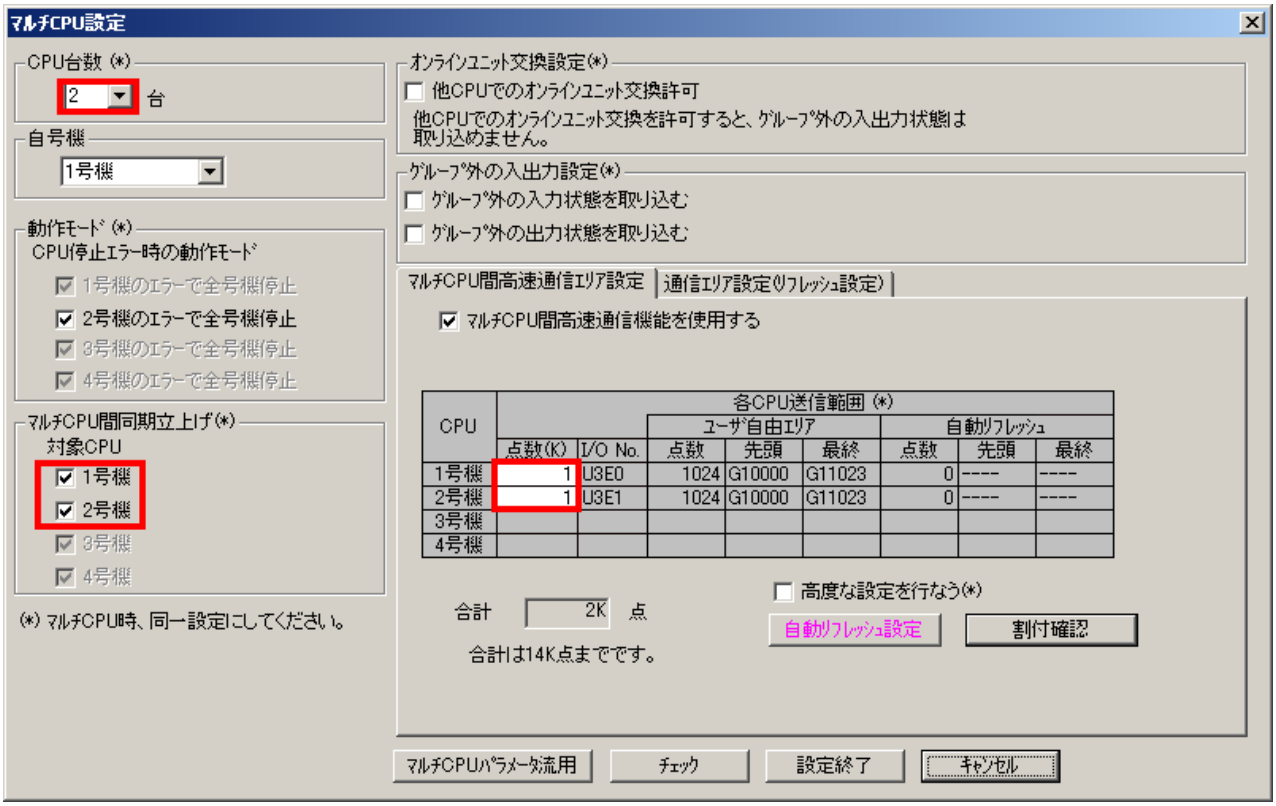


図 6-1 : シーケンサ CPU : GX Developer による設定画面例

マルチ CPU 間高速通信エリアには、ユーザ自由エリアと、自動リフレッシュエリアが設定できますが、ロボット CPU(Q172DRCPU) は、自動リフレッシュエリアに対応しておりませんので、自動リフレッシュエリアの点数は、常に 0 としてください。なお、ロボット CPU 以外の CPU の設定については、各 CPU の取扱説明書を参照ください。

※1) マルチ CPU および、マルチ CPU 間高速通信エリアについては、QCPU のマニュアル (QCPU ユーザマニュアル マルチ CPU システム編) を参照ください。

(2) ロボット CPU のパラメータ設定

RT ToolBox を用いて、マルチ CPU のパラメータ設定をおこないます。

表 6-2 : ロボット CPU のパラメータ設定

<table>
<tr><th>パラメータ名</th><th>内容</th><th>出荷時の設定値</th></tr>
<tr><td>QMLTCPUN</td><td>マルチ CPU 台数設定
マルチ CPU システムで、基本ベースユニットに装着する CPU ユニットの台数を設定します。
範囲 : 1 ～ 4</td><td>2</td></tr>
<tr><td>QMLTCPUn
n = 1 ～ 4</td><td>マルチ CPU n 号機高速通信エリア設定 (n = 1 ～ 4)
マルチ CPU システムで、1 ～ 4 号機のマルチ CPU 間高速通信機能の各 CPU ユニット間で送受信をおこなう点数を設定します。
全 CPU についてパラメータ設定値を合わせる必要があります。
パラメータ設定値が合っていない場合は、シーケンサ CPU でエラーになりますので、各 CPU のパラメータ設定値を合わせるようにしてください。

要素 1 : ユーザ自由エリアのサイズ (K 点)
範囲 : 1 ～ 14(最大)
表 6-3 : CPU 台数別の設定範囲
<table><tr><th>CPU 台数</th><th>設定範囲</th></tr><tr><td>2</td><td>0 ～ 14K 点</td></tr><tr><td>3</td><td>0 ～ 13K 点</td></tr><tr><td>4</td><td>0 ～ 12K 点</td></tr></table>
要素 2 : 自動リフレッシュの点数 (点)
範囲 : 0 ～ 14335
ただしロボット CPU は、自動リフレッシュエリアに対応しておりませんので、ロボット CPU の自動リフレッシュエリアの点数は、常に 0 としてください。
要素 3 : システムエリアのサイズ (K 点)
範囲 : 1 または 2
要素 4 : マルチ CPU 同期立ち上げ (1 : する、0 : しない)
ロボット CPU は、立ち上がりに時間がかかりますので、基本的に 1(同期する) のまま、設定を変えないでください。
全 CPU について同一設定にする必要があります。</td><td>1,0,1,1</td></tr>
<tr><td>IQMEM 注 1)</td><td>共有メモリ拡張機能を選択します。
ビット毎に機能を割り当てています。1/0= 有効 / 無効
15 0
00000000 00000000
bit2-3、5-15 は未使用
|...bit0: 拡張機能の使用
|....bit1: シーケンサダイレクト実行機能
|.....bit4: ロボット間協調制御機能</td><td>0000000000000000</td></tr>
<tr><td>IQSPEC 注 1)</td><td>CR7xx-Q シリーズコントローラの機能を設定します。
ビット毎に機能を割り当てています。1/0= 有効 / 無効
15 0
00000000 00000000
bit1-15 は未使用
|...bit0: 共有メモリ書込み方向
=0: 読み出し, 書込み共先頭から最終ｱﾄﾞﾚｽの順に実施
=1: 読み出しは先頭から, 書込みは最終ｱﾄﾞﾚｽから順に実施</td><td>0000000000000001</td></tr>
</table>

注 1）拡張機能（シーケンサダイレクト実行機能）の詳細については別冊の「取扱説明書 /CR750-Q シリーズ、CRnQ-700 シリーズ iQ Platform 対応拡張機能説明書 (BFP-A8757)」を参照願います。

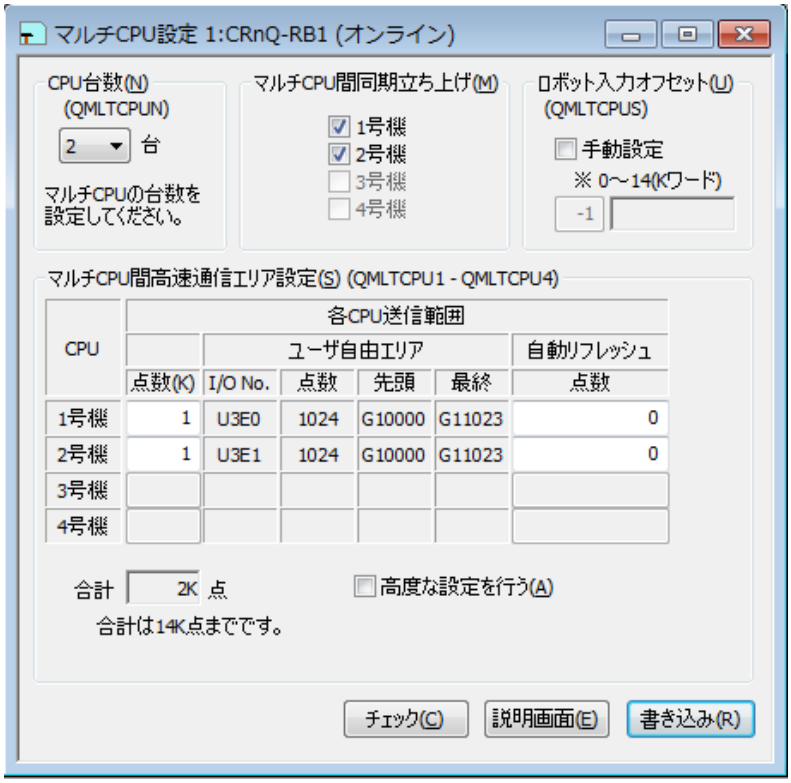


図 6-2：ロボット CPU：RT ToolBox による設定画面例

◇◆◇マルチ CPU の対象について◇◆◇

マルチ CPU の対象は、下記の iQ Platform 対応の CPU、ベースとします。(2010 年 8 月現在)

項目	型名	備考
シーケンサ CPU	ユニバーサルモデル QCPU Q03UD(E)CPU、Q04UD(E)HCPU、 Q06UD(E)HCPU、Q10UD(E)HCPU、 Q13UD(E)HCPU、Q20UD(E)HCPU、 Q26UD(E)HCPU、Q50UDEHCPU、 Q100UDEHCPU	・マルチ CPU 間高速通信に対応したもの ・1 号機はシーケンサ CPU であることが必要
ロボット CPU	Q172DRCPU	
モーション CPU	Q172DCPU、Q173DCPU	
NC CPU	Q172NCCPU	
ベース	マルチ CPU 間高速基本ベース Q38DB、Q312DB	・マルチ CPU 間高速通信に対応したもの

ロボット CPU は、RT ToolBox またはティーチングボックス (R33TB、R57TB)、シーケンサ CPU は、GX Developer、モーションコントローラ CPU は、MT Devevoper、NC CPU は、Remote monitor Tool　等を用いて、パラメータ設定します。詳細については、各ツールの取扱説明書を参照してください。

6.2.2 CPU共有メモリとロボット入出力信号の対応

シーケンサ CPU では、CPU 共有メモリを U3E0¥G10000 のようにアクセスします。ロボット CPU No.n の CPU 共有メモリは U3En¥G10000 のようにしてアクセスします 。
(n = 1 ～ 3、最大 3 台のロボット CPU が使用可能)
ロボット CPU の入出力信号番号は、各 10000 ～ 18191 となっています。
シーケンサ側はワードデバイス、ロボット側はビットデバイスとなっていますので注意が必要です。
なお、CPU 共有メモリとロボット入出力信号との対応は下表の通りになっており、変更はできません。

表 6-4 : CPU 共有メモリとロボット入出力信号の対応

シーケンサ （ワードデバイス）		ロボット （ビットデバイス）	
出力	U3E0¥G10000 ～ U3E0¥G10511	入力	ロボット CPU No.1 / 10000 ～ 18191
	U3E0¥G10512 ～ U3E0¥G11023		ロボット CPU No.2 / 10000 ～ 18191
	U3E0¥G11024 ～ U3E0¥G11535		ロボット CPU No.3 / 10000 ～ 18191
入力	U3E1¥G10000 ～ U3E1¥G10511	出力	ロボット CPU No.1 / 10000 ～ 18191
	U3E2¥G10000 ～ U3E2¥G10511		ロボット CPU No.2 / 10000 ～ 18191
	U3E3¥G10000 ～ U3E3¥G10511		ロボット CPU No.3 / 10000 ～ 18191

6.2.3 シーケンスラダーの例

操作盤の「ロボット操作権有効ボタン」X0 を ON して、操作盤の「ロボット操作権有効中ランプ」Y20 に、ロボットの操作権の有効状態を出力する例を示します。マルチ CPU 構成は、1 号機 : シーケンサ QnUD(H)CPU、 2 号機 : ロボット Q172DRCPU　です。

【解説】

<0 ～ 16 行目 >

M100 ～ M131 を U3E0¥G10000、U3E0¥G10001 の共有デバイスメモリへ書き込み、シーケンサからロボットへの入力としています。U3E1¥G10000、U3E1¥G10001 の共有デバイスメモリを、M200 ～ M231 のビットデバイスに読み出し、ロボットからシーケンサへの出力としています。

<17 ～ 22 行目 >

X0 を ON すると M105 を ON し、M105 に対応するシーケンサの U3E0¥G10000 のビット 5 が ON します。すると、ロボットの入力 10005 が ON し、専用入力信号で割り当てられている操作権が有効になります。

操作権が有効になると、専用出力信号で割り当てられているロボットの出力 10005 が ON して、ロボットの U3E1¥G10000 のビット 5 が ON します。すると、U3E1¥G10000 のビット 5 に対応するシーケンサの M205 が ON し、Y20 が ON するということになります。

なお、この例でビットデバイス M201(U3E0¥G10000 のビット 1 / つまりロボットの出力 10001) は、コントローラ電源 ON 完了 (外部入力信号が受付可能であることを出力) を表します。

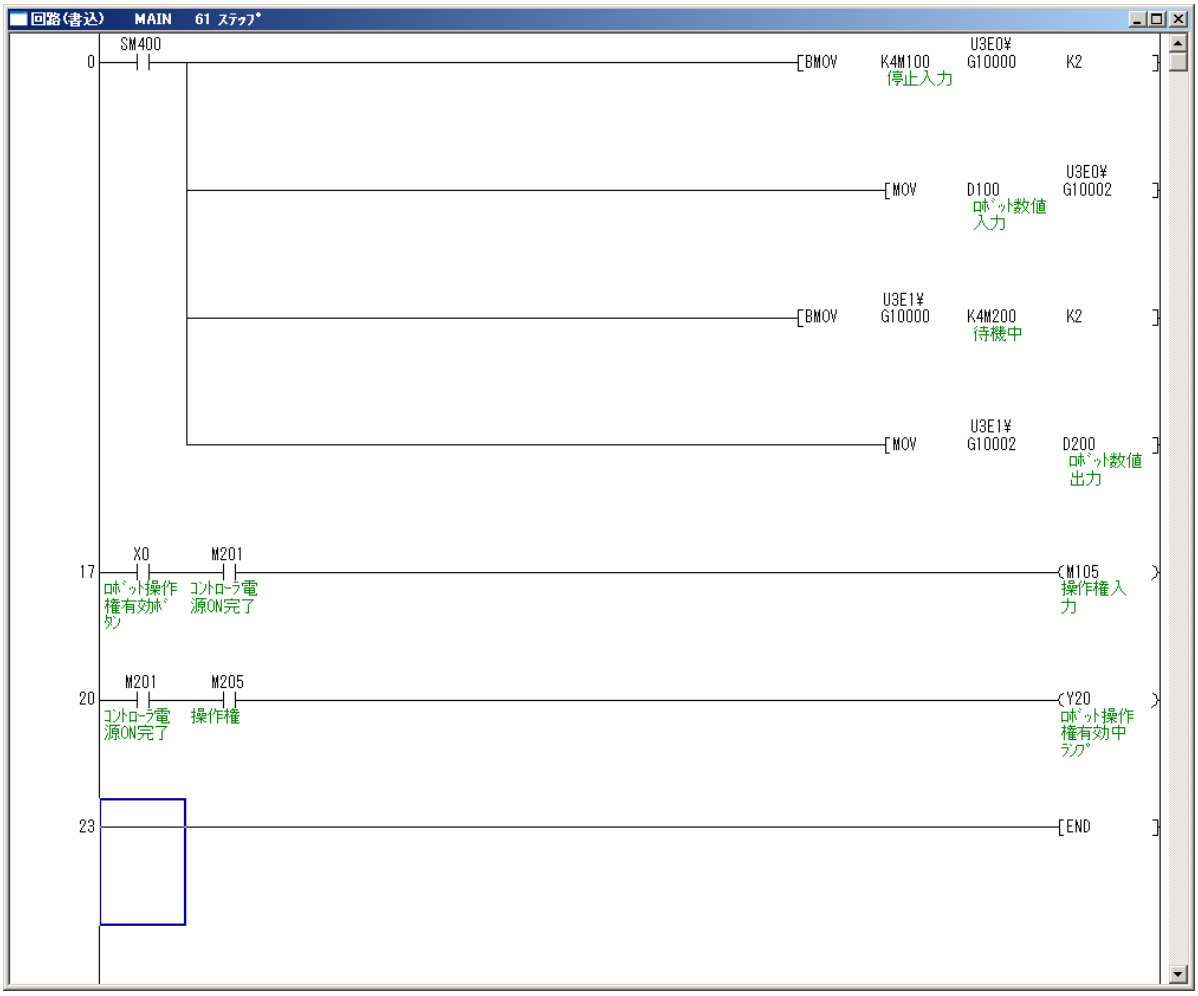


図 6-3：シーケンサラダー例

6.2.4 専用入出力信号の割付(工場出荷時設定)

工場出荷時の専用入出力信号の割付を表 6-5 に示します。

表 6-5：専用入出力信号の割付（工場出荷時設定）

パラメータ名	入力信号名称（※ 操作権必要）	出力信号名称	入力	出力	Gデバイス 注1)
STOP	停止入力（割付変更不可）	中断中出力	10000	10000	G10000
RCREADY	－	コントローラ電源 ON 完了	－	10001	
ATEXTMD	－	リモートモード出力	－	10002	
TEACHMD	－	ティーチモード出力	－	10003	
ATTOPMD	－	OP モード出力	－	10004	
IOENA	操作権入力	操作権出力	10005	10005	
START	始動入力（※）	運転中出力	10006	10006	
STOPSTS	－	停止信号入力中	－	10007	
SLOTINIT	プログラムリセット（※）	プログラム選択可能出力	10008	10008	
ERRRESET	エラーリセット入力	エラー発生中出力	10009	10009	
SRVON	サーボ ON 入力（※）	サーボ ON 中出力	10010	10010	
SRVOFF	サーボ OFF 入力	サーボ ON 不可出力	10011	10011	
CYCLE	サイクル停止入力	サイクル停止動作中出力	10012	10012	
SAFEPOS	退避点復帰信号（※）	退避点復帰中出力	10013	10013	
BATERR	－	バッテリ電圧低下	－	10014	
OUTRESET	汎用出力信号リセット（※）	－	10015	－	
HLVLERR	－	ハイレベルエラー出力	－	10016	G10001
LLVLERR	－	ローレベルエラー出力	－	10017	
CLVLERR	－	警告レベルエラー出力	－	10018	
EMGERR	－	非常停止出力	－	10019	
PRGSEL	プログラム選択入力	－	10020	－	
OVRDSEL	オーバーライド選択入力	－	10021	－	
PRGOUT	プログラム番号出力要求	プログラム番号出力中	10022	10022	
LINEOUT	行番号出力要求	行番号出力中	10023	10023	
OVRDOUT	オーバーライド値出力要求	オーバーライド値出力中	10024	10024	
ERROUT	エラー番号出力要求	エラー番号出力中	10025	10025	
－	－	－	－	－	
－	－	－	－	－	
－	－	－	－	－	
－	－	－	－	－	
－	－	－	－	－	
－	－	－	－	－	
IODATA	数値入力 0	数値出力 0	10032	10032	G10002
	数値入力 1	数値出力 1	10033	10033	
	数値入力 2	数値出力 2	10034	10034	
	数値入力 3	数値出力 3	10035	10035	

パラメータ名	入力信号名称 (※ 操作権必要)	出力信号名称	入力	出力	Gデバイス 注 1)
IODATA	数値入力 4	数値出力 4	10036	10036	G10002
	数値入力 5	数値出力 5	10037	10037	
	数値入力 6	数値出力 6	10038	10038	
	数値入力 7	数値出力 7	10039	10039	
	数値入力 8	数値出力 8	10040	10040	
	数値入力 9	数値出力 9	10041	10041	
	数値入力 10	数値出力 10	10042	10042	
	数値入力 11	数値出力 11	10043	10043	
	数値入力 12	数値出力 12	10044	10044	
	数値入力 13	数値出力 13	10045	10045	
	数値入力 14	数値出力 14	10046	10046	
	数値入力 15	数値出力 15	10047	10047	
HNDCNTL1	-	ハンド出力信号状態 900	-	10048	G10003
	-	ハンド出力信号状態 901	-	10049	
	-	ハンド出力信号状態 902	-	10050	
	-	ハンド出力信号状態 903	-	10051	
	-	ハンド出力信号状態 904	-	10052	
	-	ハンド出力信号状態 905	-	10053	
	-	ハンド出力信号状態 906	-	10054	
	-	ハンド出力信号状態 907	-	10055	
HNDSTS1	-	ハンド入力信号状態 900	-	10056	
	-	ハンド入力信号状態 901	-	10057	
	-	ハンド入力信号状態 902	-	10058	
	-	ハンド入力信号状態 903	-	10059	
	-	ハンド入力信号状態 904	-	10060	
	-	ハンド入力信号状態 905	-	10061	
	-	ハンド入力信号状態 906	-	10062	
	-	ハンド入力信号状態 907	-	10063	
USRAREA	-	ユーザ定義領域 1	-	10064	G10004
	-	ユーザ定義領域 2	-	10065	
	-	ユーザ定義領域 3	-	10066	
	-	ユーザ定義領域 4	-	10067	
	-	ユーザ定義領域 5	-	10068	
	-	ユーザ定義領域 6	-	10069	
	-	ユーザ定義領域 7	-	10070	
	-	ユーザ定義領域 8	-	10071	

注 1) マルチ CPU 間共有デバイスの番地。(シーケンサ CPU 側から見た番地)

6.3 専用入出力

専用入力出力信号には、表 6-6 に示す機能があります。
これらは、それぞれのパラメータに信号番号を割付けてパラレル入出力ユニットで使用します。
信号番号の割付けは、各パラメータに〝入力信号〟、〝出力信号〟の順に使用する信号番号を指定します。
パラメータの設定方法は 85 ページの「3.14 パラメータ画面の操作」を参照ください。なお、割り付ける信号番号に -1 を指定すると、その信号は未使用になります。

入出力のパラメータは T/B のパラメータ画面かパソコンサポートソフトウエア（オプション）のメンテナンスツールにて設定できます。また 536 ページの「6.5.2 タイミングチャート例」に信号のタイムチャートを示しますので参考にしてください。
専用入力信号は、コントローラのモードが〝AUTOMATIC〟で、操作権入力信号（IOENA）が ON している時に有効です。

表 6-6：専用入出力一覧

パラメータ名	区分	名称	機能	信号レベル	工場出荷信号番号 入力，出力	
					CR7xx-Q	CR7xx-D
RCREADY	入力	－	－		-1（無意），10001	-1（無意），-1
	出力	コントローラ電源 ON 完了	電源 ON 完了後、外部入力信号が受け付け可能であることを出力します。			
ATEXTMD	入力	－	－		-1（無意），10002	-1（無意），-1
	出力	リモートモード出力	コントローラのモードが〝AUTOMATIC〟で外部からの操作が有効であることを出力します。リモートモードのことです。 外部信号での制御のときはこの信号が ON していることが条件になります。			
TEACHMD	入力	－	－		-1（無意），10003	-1（無意），-1
	出力	ティーチモード出力	コントローラのモードが〝MANUAL〟であることを出力します。			
ATTOPMD	入力	－	－		-1（無意），10004	-1（無意），-1
	出力	自動モード出力	コントローラのモードが〝AUTOMATIC〟で操作パネルの操作が有効であることを出力します。			
IOENA	入力	操作権入力信号	外部信号制御の操作権を有効／無効にします。	レベル	10005，10005	5，3
	出力	操作権出力信号	外部信号制御の操作権有効状態を出力します。 コントローラのモードが〝AUTOMATIC〟に設定された状態で、操作権入力信号が ON しており、かつ他に操作権を取得しているデバイスが無い場合に外部信号による操作権が与えられます。			
START（操作権必要）	入力	始動入力	プログラムの始動をおこないます。指定したプログラムを始動したい場合はプログラム選択信号「PRGSEL」と数値入力「IODATA」にてプログラムを選択してから始動信号を入力してください。 ただし、パラメータ「PST」を有効にすると数値入力（IODATA）からﾌﾟﾛｸﾞﾗﾑ番号を読み込んで始動します。（プログラム選択が不要になります。） マルチタスク運転時は全タスクスロットを実行します。 但し、パラメータ「SLT**」にて起動条件が ALWAYS と ERROR 設定のスロットは除きます。	エッジ	10006，	3，
	出力	運転中出力	プログラムが運転中であることを出力します。 マルチタスク運転時は少なくとも一つのタスクスロットが運転中の時に ON します。 但し、パラメータ「SLT*」にて起動条件が ALWAYS と ERROR 設定のスロットは除きます。		10006	0

パラメータ名	区分	名称	機能	信号レベル	工場出荷信号番号 入力,出力	
					CR7xx-Q	CR7xx-D
STOP	入力	停止入力	運転中のプログラムを停止します。(起動条件がALWAYS と ERROR 設定のスロットは除きます。) 停止入力は入力信号 0 番で固定になっており、変更は不可となっています。 マルチタスク運転時は全タスクスロットを停止します。 ただし、パラメータ「SLT**」にて起動条件がALWAYS と ERROR 設定のスロットは除きます。 パラメータ INB によって A 接点 /B 接点の変更が出来ます。	レベル	10000 (変更不可),	0 (変更不可),
	出力	中断中出力	プログラムが中断中であることを出力します。 マルチタスク運転時は運転中スロットが無く、少なくとも一つのスロットが中断中の時に ON します。 ただし、パラメータ「SLT**」にて起動条件がALWAYS と ERROR 設定のスロットは除きます。		10000	-1
STOP2	入力	停止入力	運転中のプログラムを停止します。 (仕様は STOP パラメータと同じです) STOP パラメータと異なり、信号番号の変更が可能です。	レベル	-1,	-1,
	出力	中断中出力	プログラムが中断中であることを出力します。 (仕様は STOP パラメータと同じです)		-1	-1
STOPSTS	入力	-	-		-1(無意), 10007	-1(無意), -1
	出力	停止信号入力中	停止が入力中であることを出力します。(全デバイスの論理和になります)			
SLOTINIT (操作権必要)	入力	プログラムリセット	プログラムの中断中状態を解除し、実行行を先頭に戻します。プログラムリセットによってプログラム選択が可能な状態になります。 マルチタスクで使用している場合は全タスクスロットに対してプログラムリセットをおこないます。 ただし、パラメータ「SLT**」にて起動条件がALWAYS と ERROR 設定のスロットは除きます。	エッジ	10008,	-1,
	出力	プログラム選択可能出力	プログラム選択可状態であることを出力します。 プログラムが運転中 / 中断中でない場合に ON します。 マルチタスク運転時は全タスクスロットが運転中 / 中断中でない場合に ON します。 ただし、パラメータ「SLT**」にて起動条件がALWAYS と ERROR 設定のスロットは除きます。		10008	-1
ERRRESET	入力	エラーリセット入力信号	エラー状態を解除します。	エッジ	10009, 10009	2, 2
	出力	エラー発生中出力信号	エラー状態であることを出力します。			
SRVON (操作権必要)	入力	サーボ ON 入力信号	ロボットのサーボ電源を ON します。 マルチメカの場合は全メカのサーボ電源を ON します。	エッジ	10010,	4,
	出力	サーボ ON 中出力信号	ロボットのサーボ電源が ON している場合に ON になります。サーボ電源が OFF の場合は OFF になります。 マルチメカの場合は少なくとも一つのメカがサーボオン状態の時に ON します。		10010	1
SRVOFF	入力	サーボ OFF 入力信号	ロボットのサーボ電源を OFF します(全メカが対象)。 本信号が入力中はサーボオン不可になります。	レベル	10011,	1,
	出力	サーボ ON 不可出力信号	サーボ電源 ON 不可の状態を出力します。(エコーバック)		10011	-1

パラメータ名	区分	名称	機能	信号 ﾚﾍﾞﾙ	工場出荷信号番号 入力 , 出力	
					CR7xx-Q	CR7xx-D
AUTOENA	入力	自動運転可入力	非アクティブで自動運転を禁止します。 この信号が非アクティブで AUTOMATIC モードの場合、ローレベルエラー L5010 が発生します。 操作パネルでの操作と外部信号とのインターロックをとる時に使用します。必ず使用する必要はありません。	レベル	-1,	-,1
	出力	自動運転可出力	自動運転可状態を出力します。		-1	-1
CYCLE	入力	サイクル停止入力信号	サイクル停止をおこないます。	エッジ	10012, 10012	-1, -1
	出力	サイクル停止動作中出力信号	サイクル停止動作中であることを出力します。 サイクル停止が完了したら OFF します。			
MELOCK （操作権必要）	入力	マシンロック入力信号	全メカをマシンロック状態に設定 / 解除します。 全スロットがプログラム選択可状態の時に設定 / 解除が可能です。プログラム選択可状態の時はレベルになります。	レベル	-1,	-1,
	出力	マシンロック中出力信号	マシンロック状態を出力します。 少なくとも一つのメカがマシンロック状態の時に ON します。マシンロック状態の場合、ﾛﾎﾞｯﾄは動作せずにプログラム運転が可能になります。		-1	-1
SAFEPOS （操作権必要）	入力	待避点復帰入力信号	待避点復帰動作を要求します。パラメータ「JSAFE」の位置へ関節補間動作します。速度はオーバーライドに依存します。周辺装置への干渉にご注意ください。	エッジ	10013,	-1,
	出力	待避点復帰中出力信号	待避点復帰動作中であることを出力します。		10013	-1
BATERR	入力	－	－		-1(無意), 10014	-1, -1
	出力	バッテリ電圧低下	コントローラのバッテリ電圧が低下していることを出力します。バッテリ交換後、コントローラの電源を入れ直すと出力が OFF します。 下記条件でも出力します。 ・コントローラ電源 OFF 積算時間が 14600 時間を超えた。 バッテリ消耗時間を初期化すると出力が OFF します。			
OUTRESET （操作権必要）	入力	汎用出力信号リセット	汎用出力信号をリセットします。 パラメータ ORS1000 ～ ORS1816 によって入力時の動作を設定します。	エッジ	10015,	-1,
	出力	－	－		-1(無意)	-1(無意),
HLVLERR	入力	－	－		-1(無意), 10016	-1(無意), -1
	出力	ﾊｲﾚﾍﾞﾙｴﾗｰ出力信号	ハイレベルエラー発生中であることを出力します。			
LLVLERR	入力	－	－		-1(無意), 10017	-1(無意), -1
	出力	ﾛｰﾚﾍﾞﾙｴﾗｰ出力信号	ローレベルエラー発生中であることを出力します。			
CLVLERR	入力	－	－		-1(無意), 10018	-1(無意), -1
	出力	警告ﾚﾍﾞﾙｴﾗｰ出力信号	警告レベルエラー発生中であることを出力します。			
EMGERR	入力	－	－		-1(無意), 10019	-1(無意), -1
	出力	非常停止出力信号	非常停止が発生中であることを出力します。 [EMGERR 出力条件] ・外部非常停止エラー：H0050、H0051 （二重系ライン異常） ・操作パネル非常停止エラー：H0060、H0061 （二重系ライン異常） ・T/B 非常停止エラー：H0070、H0071 （二重系ライン異常） ・ドアスイッチエラー：H0039、H0040 （二重系ライン異常） ・EMGIN コネクタの配線異常：H0141			

パラメータ名	区分	名称	機能	信号 ﾚﾍﾞﾙ	工場出荷信号番号 入力,出力	
					CR7xx-Q	CR7xx-D
SnSTART (n=1 ～ 32) (操作権必要)	入力	ｽﾛｯﾄ n 始動入力	各スロットの始動をおこないます。n=1 ～ 32	エッジ	-1, -1	-1, -1
	出力	ｽﾛｯﾄ n 運転中出力	各スロットの運転中状態を出力します。n=1 ～ 32			
SnSTOP (n=1 ～ 32)	入力	ｽﾛｯﾄ n 停止入力	各スロットの停止をおこないます。n=1 ～ 32	レベル	-1, -1	-1, -1
	出力	ｽﾛｯﾄ n 中断中出力	各スロットが一時停止中で、プログラム運転が中断中であることを出力します。n=1 ～ 32			
MnSRVOFF (n=1 ～ 3)	入力	ﾒｶ n ｻｰﾎﾞ OFF 入力信号	各メカをサーボオフします。n=1 ～ 3 本信号が入力中はサーボオン不可になります	レベル	-1,	-1,
	出力	ﾒｶ n ｻｰﾎﾞ ON 不可出力信号	サーボオン不可状態を出力します。 (エコーバック)		-1	-1
MnSRVON (n=1 ～ 3) (操作権必要)	入力	ﾒｶ n ｻｰﾎﾞ ON 入力信号	各メカをサーボオンします。n=1 ～ 3	エッジ	-1, -1	-1, -1
	出力	ﾒｶ n ｻｰﾎﾞ ON 中出力信号	サーボオン状態を出力します。n=1 ～ 3			
MnMELOCK (n=1 ～ 3) (操作権必要)	入力	ﾒｶ n ﾏｼﾝﾛｯｸ入力信号	各メカをマシンロック状態に設定 / 解除します。 n=1 ～ 3	エッジ	-1, -1	-1, -1
	出力	ﾒｶ n ﾏｼﾝﾛｯｸ中出力信号	マシンロック状態であることを出力します。 n=1 ～ 3			
PRGSEL (操作権必要)	入力	プログラム選択入力信号	数値入力信号での設定値をプログラム番号に指定します。 スロット 1 に対してプログラムを選択します。 数値入力 (IODATA) への出力が 15ms 以上たってから出力してください。また本信号も 15ms 以上はロボットへ出力してください。	エッジ	10020	-1,
	出力	-	-			
OVRDSEL (操作権必要)	入力	ｵｰﾊﾞｰﾗｲﾄﾞ選択入力信号	数値入力信号での設定値をｵｰﾊﾞｰﾗｲﾄﾞとします。 数値入力 (IODATA) への出力が 15ms 以上たってから出力してください。また本信号も 15ms 以上はロボットへ出力してください。	エッジ	10021	-1,
	出力	-	-			
IODATA	入力	数値入力 (開始ビット番号, 終了ビット番号)	バイナリー値として読み込みます。 ・プログラム番号 (PRGSEL にて読み込みます)、パラメータ「PST」有効時は始動信号にて読み込みます。 ・ｵｰﾊﾞｰﾗｲﾄﾞ (OVRDSEL にて読み込みます)、 ビット幅は任意に設定できます。ただし、設定したビット幅を超える値を出力した場合は値は保証されません。 ロボットへの入力は最低 15ms 以上出力してから PRGSEL などの設定信号を入力してください。	レベル	注 2) 10032 (開始ﾋﾞｯﾄ), 10047 (終了ﾋﾞｯﾄ),	注 2) -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ),
	出力	数値出力 (開始ビット番号, 終了ビット番号)	バイナリー値として出力します。 ・プログラム番号 (PRGOUT にて出力します)、 ・ｵｰﾊﾞｰﾗｲﾄﾞ (OVRDOUT にて出力します)、 ・行番号 (LINEOUT にて出力します)、 ・ｴﾗｰ番号 (ERROUT にて出力します) の出力を行ないます。 ビット幅は任意に設定できます。ただし、設定したビット幅を超える値を出力した場合は値は保証されません。 プログラム番号 (PRGOUT) 信号などをロボットへ入力してから 15ms 以上待ってから読込むようにしてください。		10032 (開始ﾋﾞｯﾄ), 10047 (終了ﾋﾞｯﾄ)	-1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ),
PRGOUT	入力	プログラム番号出力要求	タスクスロット 1 のプログラム番号を数値出力 (IODATA) に出力します。 本信号をロボットへ入力してから 15ms 以上待ってから数値出力 (IODATA) 信号を読み込むようにしてください。	エッジ	10022,	-1,
	出力	プログラム番号出力中信号	数値出力に「プログラム番号出力中」状態を出力します。		10022	-1

パラメータ名	区分	名称	機能	信号 ﾚﾍﾞﾙ	工場出荷信号番号 入力 , 出力	
					CR7xx-Q	CR7xx-D
LINEOUT	入力	行番号出力要求	タスクスロット 1 の行番号を数値出力 (IODATA) に出力します。 本信号をロボットへ入力してから 15ms 以上待ってから数値出力 (IODATA) 信号を読み込むようにしてください。	エッジ	10023,	-1,
	出力	行番号出力中信号	数値出力に「行番号を出力中」状態を出力します。		10023	-1
OVRDOUT	入力	ｵｰﾊﾞｰﾗｲﾄﾞ値出力要求	OP ｵｰﾊﾞｰﾗｲﾄﾞを数値出力 (IODATA) に出力します。 本信号をロボットへ入力してから 15ms 以上待ってから数値出力 (IODATA) 信号を読み込むようにしてください。	エッジ	10024,	-1,
	出力	ｵｰﾊﾞｰﾗｲﾄﾞ値出力中信号	数値出力に「ｵｰﾊﾞｰﾗｲﾄﾞを出力中」状態を出力します。		10024	-1
ERROUT	入力	エラー番号出力要求	エラー番号を数値出力 (IODATA) に出力します。 本信号をロボットへ入力してから 15ms 以上待ってから数値出力 (IODATA) 信号を読み込むようにしてください。	エッジ	10025,	-1
	出力	エラー番号出力中信号	数値出力「にエラー番号を出力中」状態を出力します。		10025	-1
JOGENA (操作権必要)	入力	ジョグ有効入力信号	指定軸を指定されたモードでジョグ動作を行ないます。 この信号が ON している間動作します。	レベル	-1,	-1,
	出力	ジョグ有効中出力信号	ジョグ動作状態であることを出力します。		-1	-1
JOGM	入力	ジョグモード入力 (開始番号, 終了番号)	ジョグモードを指定します。 0/1/2/3/4/5= 関節 / 直交 / 円筒 /3 軸直交 / ツール / ワーク (Ex-T) 注) EX-T 制御および Ex-T ジョグについては、 652 ページの「7.5Ex-T 制御について」を参照してください。	レベル	注 3) -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ), -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)	注 3) -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ), -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)
	出力	ジョグモード出力 (開始番号, 終了番号)	現在のジョグモードを出力します。			
JOGMENO ※ 以下の S/W Ver でのみ有効。 S/W Ver. F-Q ｼﾘｰｽﾞ : R5 以降 F-D ｼﾘｰｽﾞ : S5 以降	入力	ジョグメカ番号入力 (開始番号, 終了番号)	メカ番号を指定します。 本パラメータを設定していない場合は、メカ 1 固定です。	レベル	-1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ),	-1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ),
	出力	ジョグメカ番号出力 (開始番号, 終了番号)	現在のメカ番号を出力します。		-1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)	-1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)
JOG+	入力	ジョグ送りﾌﾟﾗｽ側 8 軸分 (開始番号, 終了番号)	ジョグ動作の軸を指定します。 関節ジョグ時 : 開始番号から J1,J2,J3,J4,J5,J6,J7,J8 軸 直交ジョグ時 : 開始番号から X,Y,Z,A,B,C,L1,L2 軸 円筒ジョグ時 : 開始番号から X,θ,Z,A,B,C,L1,L2 軸 三軸直交ジョグ時 : 開始番号から X,Y,Z,J4,J5,J6 軸 ツールジョグ時 : 開始番号から X,Y,Z,A,B,C 軸 ワークジョグ時 (Ex-T ジョグ時) : 開始番号から X,Y,Z,A,B,C 軸 注) EX-T 制御および Ex-T ジョグについては、 652 ページの「7.5Ex-T 制御について」を参照してください。	レベル	注 4) -1, -1	注 4) -1, -1
	出力	-	-			

パラメータ名	区分	名称	機能	信号レベル	工場出荷信号番号 入力，出力	
					CR7xx-Q	CR7xx-D
JOG-	入力	ジョグ送りﾏｲﾅｽ側 8 軸分 (開始番号，終了番号)	ジョグ動作の軸を指定します。 関節ジョグ時：開始番号から J1,J2,J3,J4,J5,J6,J7,J8 軸 直交ジョグ時：開始番号から X,Y,Z,A,B,C,L1,L2 軸 円筒ジョグ時：開始番号から X,θ,Z,A,B,C,L1,L2 軸 三軸直交ジョグ時：開始番号から X,Y,Z,J4,J5,J6 軸 ツールジョグ時：開始番号から X,Y,Z,A,B,C 軸 ワークジョグ時 (Ex-T ジョグ時)：開始番号から X,Y,Z,A,B,C 軸 注）EX-T 制御および Ex-T ジョグについては、652 ページの「7.5Ex-T 制御について」を参照してください。	レベル	注 4) -1, -1	注 4) -1, -1
	出力	-	-			
JOGWKNO	入力	ワーク座標番号	ワークジョグ動作で、基準とするワーク座標番号 (Ex-T 座標番号) を 1 から 8 の数値で指定します。 注) 本入力信号は、ジョグ有効入力信号 :JOGENA のエッジ (OFF から ON への変化) によって取込みます。ワーク座標番号を変更する場合は、ジョグ有効入力信号 :JOGENA を一旦 OFF から ON に変化させてください。 注）EX-T 制御および Ex-T 座標番号については、652 ページの「7.5Ex-T 制御について」を参照してください。	レベル	注 3) -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ), -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)	注 3) -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ), -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)
	出力		入力されているワーク座標番号 (Ex-T 座標番号) を出力します。 注）EX-T 制御および Ex-T 座標番号については、652 ページの「7.5Ex-T 制御について」を参照してください。			
JOGNER (操作権必要)	入力	ジョグ時エラー一時無視入力信号	ジョグ操作時にﾘｾｯﾄできないエラーを一時的に無視する。 ※ 本信号はメカ 1 のみ対象とする。	レベル	-1,	-1,
	出力	ジョグ時エラー一時無視中出力信号	エラー一時無視状態であることを出力する。 ※ 本信号はメカ 1 のみ対象とする。		-1	-1
HNDCNTLn (n=1 ～ 3)	入力	-	-			
	出力	メカ n ハンド出力信号状態 (開始番号，終了番号)	ハンド出力 (n=1)900 ～ 907 の状態を出力 ハンド出力 (n=2)910 ～ 917 の状態を出力 ハンド出力 (n=3)920 ～ 927 の状態を出力 例)900 ～ 903 の 4 点を汎用出力信号 3 番、4 番、5 番、6 番に出力したい場合は、HNDCNTL1 を (3, 6) に設定します。		HNDCNTL1 10048 (開始ﾋﾞｯﾄ), 10055 (終了ﾋﾞｯﾄ)	-1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)
HNDSTSn (n=1 ～ 3)	入力	-	-			
	出力	メカ n ハンド入力信号状態 (開始番号，終了番号)	ハンド入力 (n=1)900 ～ 907 の状態を出力 ハンド入力 (n=2)910 ～ 917 の状態を出力 ハンド入力 (n=3)920 ～ 927 の状態を出力 例)900 ～ 903 の 4 点を汎用出力信号 3 番、4 番、5 番、6 番に出力したい場合は、HNDSTS1 を (3, 6) に設定します。		HNDSTS1 10056 (開始ﾋﾞｯﾄ), 10063 (終了ﾋﾞｯﾄ)	-1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)
HANDENA	入力	ハンド制御許可入力	外部信号によるロボットハンドの制御を許可、または禁止します。 1/0 = 許可 / 禁止 注) 自動運転中にロボットのハンド制御が可能となります。安全確保のためロボットとシーケンサ等の外部機器とのインターロックを確実におこなってください。 外部信号によるロボットハンド制御を許可している場合、プログラムからの「HOpen/ HClose」は無効になります。	レベル	-1, -1	-1, -1
	出力	ハンド制御許可出力	外部信号によるロボットハンドの制御の許可状態を出力します。 1/0 = 許可 / 禁止 ハンド制御許可入力信号が ON されている時、T/B が有効でない場合に本信号が ON します。			

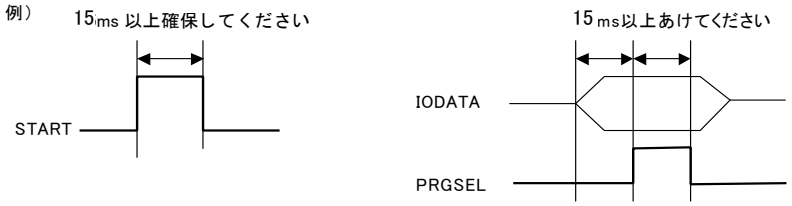
パラメータ名	区分	名称	機能	信号レベル	工場出荷信号番号 入力 , 出力	
					CR7xx-Q	CR7xx-D
HANDOUT	入力	ハンド出力制御信号	ロボットハンドを制御するための外部入力信号範囲を設定します。 ここで設定した入力信号は、 パラメータ : HANDTYPE で設定されたハンド信号に順番に対応付けられます。 要素 1 : ハンド出力制御信号開始番号 要素 2 : ハンド出力制御信号終了番号	エッジ	-1, -1	-1, -1
	出力	-	-			
HNDERRn (n=1 ～ 3)	入力	メカ n ハンドエラー入力信号	ハンドエラー発生を要求します。 ローレベルエラー 30 番が発生します。	レベル	-1,	-1,
	出力	メカ n ハンドエラー中出力信号	ハンドエラー発生中状態を出力します。		-1	-1
AIRERRn (n=1 ～ 5)	入力	メカ n 空気圧ｴﾗｰ入力信号	空圧エラー発生を要求します。 ローレベルエラー 31 番が発生します。	レベル	-1,	-1,
	出力	メカ n 空気圧ｴﾗｰ出力中 信号	空気圧エラー発生中状態を出力します。		-1	-1
USRAREA 435 ページの「5.8 ユーザ定義領域について」してください。	入力	-	-		注 5) 10064 (開始ﾋﾞｯﾄ), 10071 (終了ﾋﾞｯﾄ)	注 5) -1(開始ﾋﾞｯﾄ), -1(終了ﾋﾞｯﾄ)
	出力	ユーザ定義領域 32 点 (開始番号 , 終了番号)	ロボットがユーザ定義領域内にいることを出力します。 開始番号側から領域 1,2,3 の順に出力されます。 領域はﾊﾟﾗﾒｰﾀ、AREA1P1、AREA1P2 ～ AREA32P1、AREA32P2 で設定します。 設定例) USRAREA を例にすると 領域 1 のみ使用の場合 USRAREA : 8,8 → 設定有効 領域 1,2 を使用の場合 USRAREA : 8,9 → 設定有効 USRAREA : -1,-1 → 設定無効 USRAREA : 8,-1 → 設定無効 (エラーなし) USRAREA : -1,8 → 設定無効 (エラーなし) USRAREA : 9,8 → 設定無効 (L6643 エラー)			
MnPTEXC (n=1 ～ 3)	入力	-	-		-1(無意),	-1(無意),
	出力	メンテ品の交換時期警告	メンテナンス品の交換時期に達していることを出力します。	レベル	-1	-1
MnWUPENA (n=1 ～ 3) (操作権必要)	入力	メカ n 暖機運転モード有効入力信号	各メカの暖機運転モードを有効にします。 (n=1 ～ 3) 注) 本入力信号で暖機運転モードの有効 / 無効を切り替えるには、あらかじめパラメータ (WUPENA 等) により暖機運転モードを有効にしておく必要があります。パラメータにより暖機運転モードが無効となっている場合は、本入力信号を入力しても有効にはなりません。	レベル	-1,	-1,
	出力	メカ n 暖機運転モード出力信号	暖機運転モードが有効となっていることを出力します。 (n=1 ～ 3)		-1	-1
MnWUPMD (n=1 ～ 3)	入力	-	-		-1(無意),	-1(無意),
	出力	メカ n 暖機運転状態出力信号	暖機運転状態であり速度を下げて動作することを出力します。(n=1 ～ 3)		-1	-1
PSSLOT	入力	位置データ出力スロット番号指定	位置データの出力対象とするスロット番号 (1 ～ 32) を指定します。ここで指定するスロットに予め対象プログラムをロードしておきます。 * 外部機器側でパラメータ : PSOUT の入力信号 (位置データ出力指示) が OFF していないとスロット番号は変更されません。	レベル	-1 (入力開始ﾋﾞｯﾄ), -1 (入力終了ﾋﾞｯﾄ), -1 (出力開始ﾋﾞｯﾄ), -1 (出力終了ﾋﾞｯﾄ) * 入出力共に最大 6bit 幅	
	出力	位置データ出力スロット番号出力	現在位置データの出力対象となっているスロット番号を出力 (応答) します。			

パラメータ名	区分	名称	機能	信号レベル	工場出荷信号番号 入力,出力	
					CR7xx-Q	CR7xx-D
PSTYPE	入力	位置データタイプ指定	位置データの出力対象とするデータの型を指定します。 1/0= 関節型変数 / 位置型変数 ＊外部機器側でパラメータ：PSOUT の入力信号（位置データ出力指示）が OFF していないと位置データタイプは変更されません。	レベル	-1 , -1	
	出力	位置データタイプ出力	現在位置データの出力対象となっているデータ型を出力（応答）します。 1/0= 関節型変数 / 位置型変数			
PSNUM	入力	位置データ番号指定	出力対象とする位置データの番号（関節型変数 / 位置型変数の番号）を指定します。 位置データの番号範囲：0 ～ 65535 例)P100 を出力対象とする場合、入力開始ﾋﾞｯﾄ～入力終了ﾋﾞｯﾄ間に数値の ″100″ を指定します。 （注)″P001″ のように上位の桁に ″0″ を含んでいる変数は指定できません。 ＊外部機器側で PSOUT 入力信号（位置データ出力指示）が OFF していないと位置データ番号は変更されません。	レベル	-1（入力開始ﾋﾞｯﾄ）， -1（入力終了ﾋﾞｯﾄ）， -1（出力開始ﾋﾞｯﾄ）， -1（出力終了ﾋﾞｯﾄ） ＊入出力共に最大 16bit 幅	
	出力	位置データ番号出力	現在出力対象となっている位置データの番号を出力（応答）します。 入力側で指定されたポジション番号を出力します。			
PSOUT	入力	位置データ出力指示	現在出力対象となっている位置データを出力させる指示です。 この信号がオフからオンに変化した時（エッジ）に、現在出力対象となっている位置データをパラメータ：PSPOS（位置データ出力ビット領域）で指定されている信号番号へ出力し、この信号が継続してオンの間（レベル）、その出力状態を維持します。（リアルタイム更新はおこないません） （エッジ：位置データの出力更新、レベル：出力値維持） 0 → 1：位置データの出力更新 1：出力状態維持 0：出力対象位置データ変更可能	エッジ + レベル	-1 , -1	
	出力	位置データ出力中	指定された位置データを出力していることを出力します。 【出力情報】 OFF：位置データ未出力 ON ：位置データ出力中			

パラメータ名	区分	名称	機能	信号レベル	工場出荷信号番号 入力 , 出力	
					CR7xx-Q	CR7xx-D
PSPOS	入力	-	-	-	-1 , (無意)	
	出力	位置データ出力信号領域	位置データを出力する信号領域の先頭番号を指定します。 領域は 8 軸分の座標値 (32bit x 8 軸) と構造フラグ (32bit x 2 要素) で、計 320 ビットの連続領域が必要です。以下の範囲が設定できます。 (1)CR7xx-Q シリーズ 10000 ～ 17872 : マルチ CPU 間共有デバイス (2)CR7xx-D シリーズ 2000 ～ 3632 : プロフィバス 6000 ～ 7728 : CC-Link 出力する位置データの並び 位置型変数 : X,Y,Z,A,B,C,L1,L2,FL1,FL2 関節型変数 : J1,J2,J3,J4,J5,J6,J7,J8 各座標値は 1000 倍した整数で出力します。(mm, 度) 外部機器側では 1/1000 倍して扱ってください。但し、FL1,FL2 はそのままの値を出力します。 出力できるデータ範囲 符号付整数 -2147483648 ～ 2147483647 540 ページの「(5) 外部信号操作タイムチャート (その 5)」にタイムチャートと注意事項を示しますので参照願います。		-1 -1	
TMPOUT	入力	盤内温度出力要求	ﾛﾎﾞｯﾄｺﾝﾄﾛｰﾗの盤内温度を数値出力 (IODATA) に出力します。 温度は整数で, ﾊﾞｲﾅﾘ値 (2 進数) で出力します。 本信号をﾛﾎﾞｯﾄへ入力してから 15ms 以上待ってから数値出力 (IODATA) 信号を読み込むようにしてください。	エッジ	-1 , -1	
	出力	盤内温度出力中	数値出力に「盤内温度を出力中」状態を出力します。	-		
RSTBAT	入力	バッテリ積算時間のリセット	バッテリ積算時間をリセットします。	-	-1、	
	出力	リセット完了	リセット完了を出力します。		-1	
RSTGRS	入力	メンテ予報のリセット (グリス)	メンテ予報のグリス情報をリセットします。 ※ 軸ビットパターンは IODATA または DIODATA で指定します。	-	-1、	
	出力	リセット完了	リセット完了を出力します。		-1	
RSTBLT	入力	メンテ予報のリセット (ベルト)	メンテ予報のベルト情報をリセットします。 ※ 軸ビットパターンは IODATA または DIODATA で指定します。	-	-1、	
	出力	リセット完了	リセット完了を出力します。		-1	
SVDATA	入力	-	-	-	-1(無意)	-1, -1,
	出力	負荷率データ	本パラメータの第 3 要素に指定したレジスタ番号から第 4 要素に指定した番号に、J1 から J8 軸の最大負荷率 (%) を出力します。 出力する負荷率の更新周期は、2 秒間隔です。 (CC-Link レジスタ対応)			-1(開始レジスタ), -1(終了レジスタ)
DOORSTS1	入力	-	-	-	-1,	-1,
	出力	ドアスイッチ 1 状態	ドアスイッチ 1 系統目の状態を出力します。		-1	-1
DOORSTS2	入力	-	-	-	-1,	-1,
	出力	ドアスイッチ 2 状態	ドアスイッチ 2 系統目の状態を出力します。		-1	-1

パラメータ名	区分	名称	機能	信号 ﾚﾍﾞﾙ	工場出荷信号番号 入力，出力	
					CR7xx-Q	CR7xx-D
DOORSTS	入力	−	−	−	−1,	−1,
	出力	ドアスイッチ状態	ドアスイッチ 1 系統目と 2 系統目の論理和を出力します。 1、2 系統目のどちらも ON の場合にのみ、この信号も ON します。		−1	−1

注 1) 信号レベルの意味を以下に示します。
レベル：→ 信号が ON 状態の時に指定された機能が有効となり、OFF 状態の時は無効となります。
信号の ON の時間は最低 15ms 以上を確保してください。
エッジ：→ 信号が OFF 状態から ON 状態に変化した時に指定された機能が有効となり、その後信号が OFF 状態に戻っても指定された機能は元の状態を維持します。



注 2) 入力開始番号、入力終了番号、出力開始番号、出力終了番号の順です。実際の数値入力または出力として使用する時は、開始番号から終了番号までを使用し 2 進数で表現します。開始番号が最下位ビット、終了番号が最上位ビットを表します。数値を表現するのに必要な数だけ設定してください。
例えばプログラム選択に使用し、プログラムが 1 ～ 6 までしかない場合は、3 ビットを設定すれば表現できます。最大 16 ビットまで設定が可能です。

以下に割付例を示します。
例) 始動入力信号を汎用入力 10016 に、運転中出力信号を汎用出力 10026 に設定する場合。
パラメータ START= {10016, 10026}
例) 数値入力を汎用入力 10027 ～ 10030 に 4 ビット分、数値出力を汎用出力 10027 ～ 10031 に 5 ビット分設定する場合。
パラメータ IODATA= {10027, 10030, 10027, 10031}

注 3) 入力開始番号、入力終了番号、出力開始番号、出力終了番号の順です。
数値は、開始番号から終了番号までを使用し 2 進数で表現します。開始番号が最下位ビット、終了番号が最上位ビットを表します。数値を表現するのに必要な数だけ設定してください。
例えば、ジョグモード入力：JOGM で関節モードと直交モードしか使用しない場合は、1 ビットを設定すれば良いことになります。

注 4) 入力開始番号，入力終了番号の順です。開始番号が J1/X 軸、終了番号が最大で J8/L2 軸を指定します。例えば 6 軸ロボットを使用した場合は、6 ビットを設定すれば良いことになります。
4 軸ロボットの場合でも、直交モードを使用する場合は、C 軸が必要になるため、6 ビットの設定が必要になります。最大 8 ビットまで設定が可能です。

注 5) 出力開始番号，出力終了番号の順です。開始番号がエリア 1、終了番号が最大でエリア 32 を指定します。例えばエリアを 2 つしか使用しない場合は、2 ビットを設定すれば良いことになります。最大 32 ビットまで設定可能です。

注 6) 上記パラメータに設定できる値の範囲は、0 ～ 255、2000 ～ 3951、6000 ～ 8047、10000 ～ 18191 です。

6.4 信号の有効無効状態

入力信号の種類によっては運転中や停止入力中など、その時のロボットの状態によっては目的とする信号を入力しても機能しない場合があります。
表 6-7 に入力信号の有効 / 無効に対するロボットの状態の関係を示します。

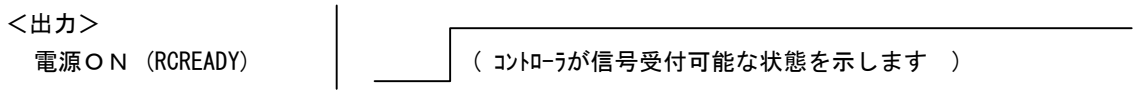
表 6-7：専用入力信号の有効 / 無効状態

パラメータ名	名称	ロボットの状態による左記信号の有効 / 無効
SLOTINIT	プログラムリセット	運転中状態 (START 出力 ON 状態) のときは機能しません。
SAFEPOS	待避点復帰入力	
OUTRESET	汎用出力信号リセット	
MnWUPENA	暖機運転モード有効入力	
START SnSTART (n=1 ～ 32)	始動入力	外部入出力の操作権有効 (IOENA 出力 ON 状態) のときのみ機能します。
SLOTINIT	プログラムリセット	
SRVON MnSRVON (n=1 ～ 3)	サーボ ON 入力	
MELOCK MnMELOCK (n=1 ～ 3)	マシンロック入力	
SAFEPOS	待避点復帰入力	
PRGSEL	プログラム選択入力	
OVRDSEL	オーバーライド選択入力	
JOGENA	ジョグ有効入力	
MnWUPENA	暖機運転モード有効入力	
START	始動入力	停止入力状態 (STOPSTS が ON 状態) のときは機能しません。
SAFEPOS	待避点復帰入力	
JOGENA	ジョグ有効入力	
SRVON	サーボ ON 入力	サーボオフ入力状態のときは機能しません。
MELOCK MnMELOCK (n=1 ～ 3)	マシンロック入力	プログラム選択可状態 (SLOTINIT 出力 ON 状態) のときのみ機能します。
PRGSEL	プログラム選択入力	中断中状態 (STOP 出力 ON 状態) のときは機能しません。

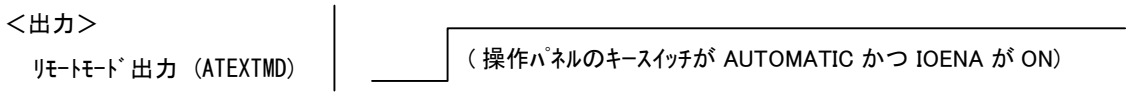
6.5 外部信号のタイミングチャート

6.5.1 各信号の個別タイミングチャート

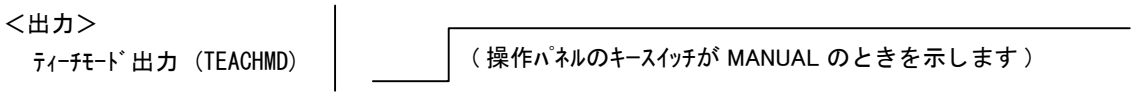
(1) RCREADY (コントローラの電源 ON 完了出力)



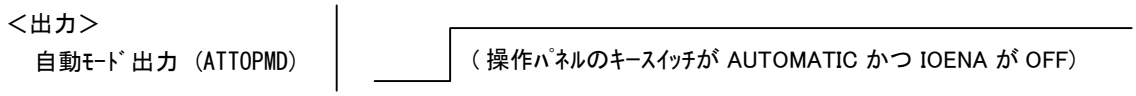
(2) ATEXTMD(リモートモード出力)



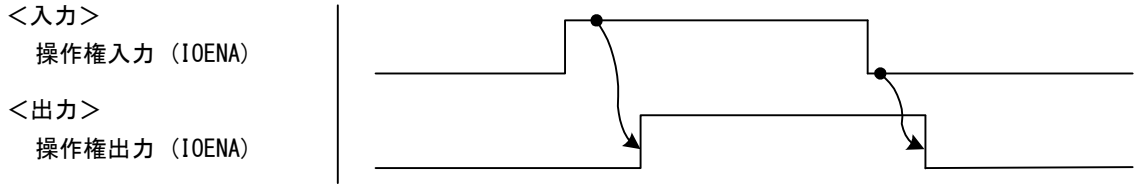
(3) TEACHMD(ティーチモード出力)



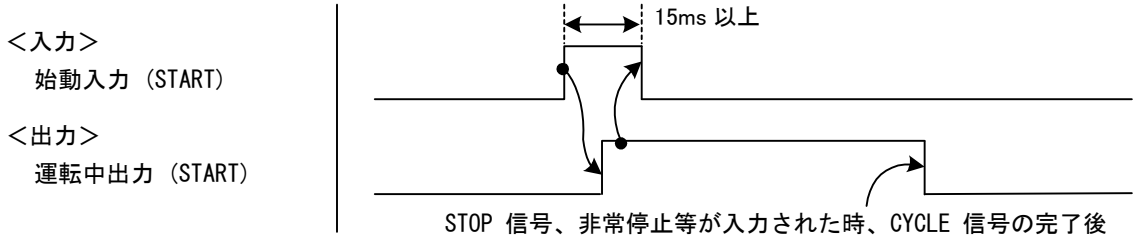
(4) ATTOPMD(自動モード出力)



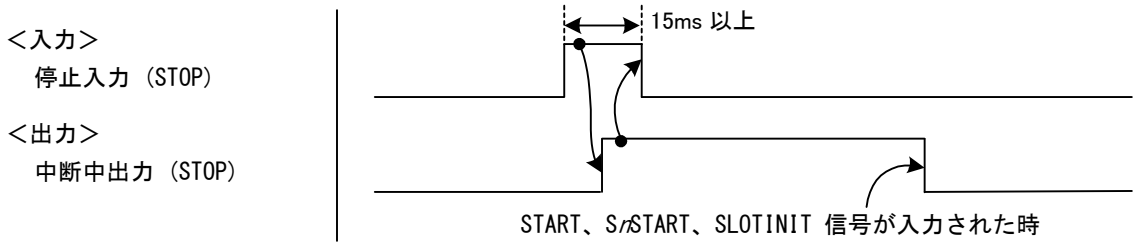
(5) IOENA (操作権入力信号 / 操作権出力信号)



(6) START (始動入力 / 運転中出力)

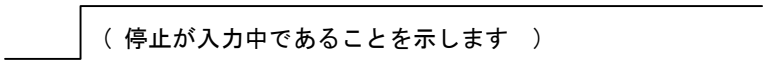


(7) STOP (停止入力 / 中断中出力)



(8) STOPSTS (停止信号入力中出力)

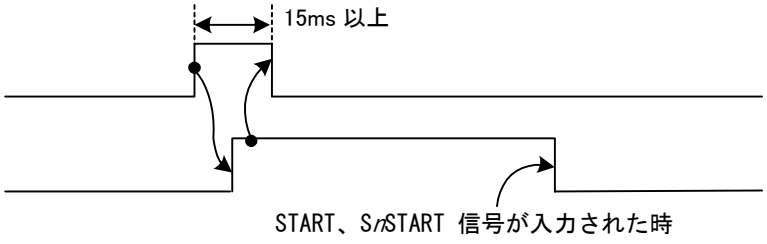
<出力>
停止信号入力中 (STOPSTS)



(9) SLOTINIT (プログラムリセット入力 / プログラム選択可能出力)

<入力>
ﾌﾟﾛｸﾞﾗﾑﾘｾｯﾄ (SLOTINIT)

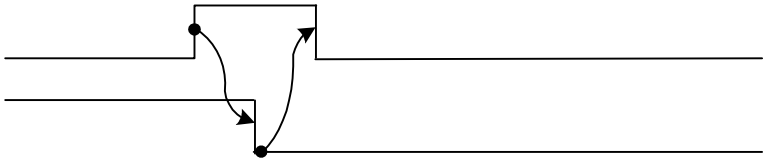
<出力>
ﾌﾟﾛｸﾞﾗﾑ選択可能出力
(SLOTINIT)



(10) ERRRESET(エラーリセット入力 / エラー発生中出力)

<入力>
ｴﾗｰﾘｾｯﾄ入力 (ERRRESET)

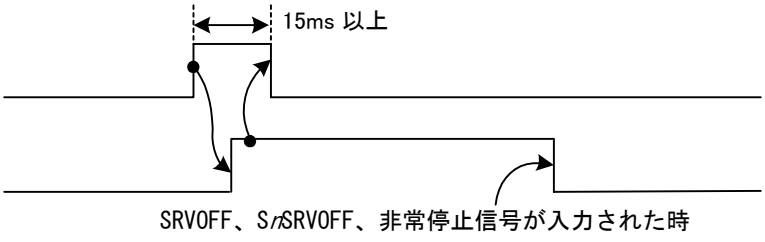
<出力>
ｴﾗｰ発生中出力 (ERRRESET)



(11) SRVON (サーボ ON 入力 / サーボ ON 中出力)

<入力>
サーボON 入力 (SRVON)

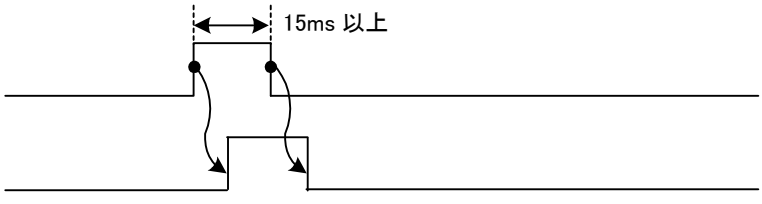
<出力>
サーボON中 出力 (SRVON)



(12) SRVOFF (サーボ OFF 入力 / サーボ ON 不可出力)

<入力>
サーボOFF 入力 (SRVOFF)

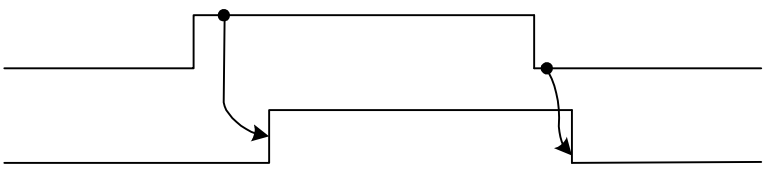
<出力>
サーボON不可 出力
(SRVOFF)



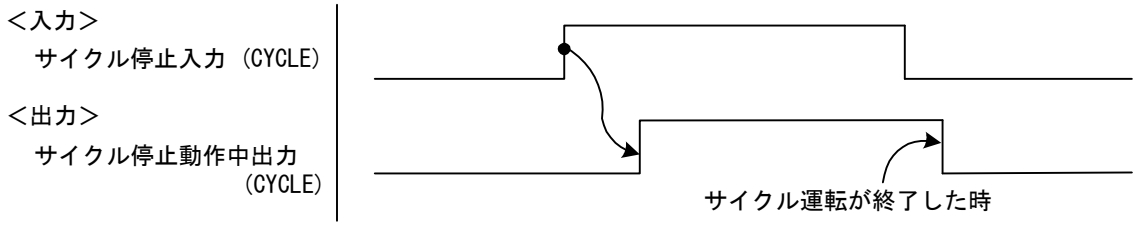
(13) AUTOENA (自動運転入力 / 自動運転可出力)

<入力>
自動運転可入力 (AUTOENA)

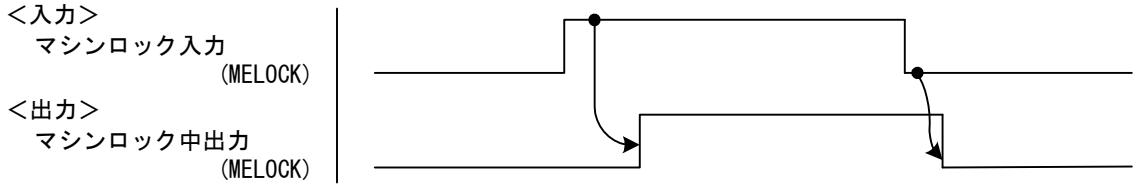
<出力>
自動運転可出力 (AUTOENA)



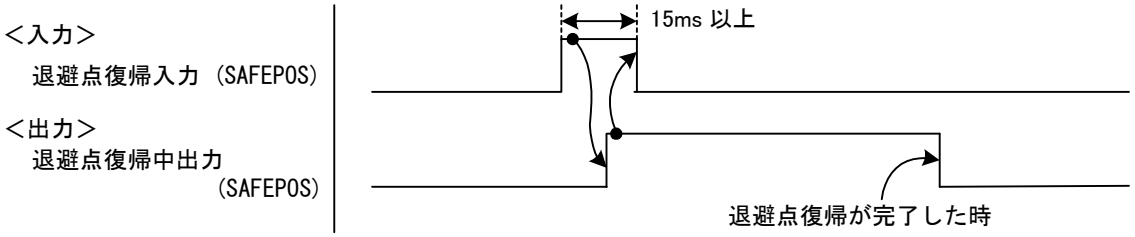
(14) CYCLE (サイクル停止入力 / サイクル停止動作中出力)



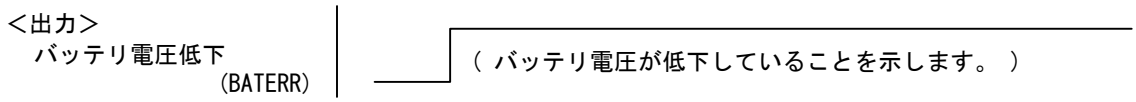
(15) MELOCK (マシンロック入力 / マシンロック中出力)



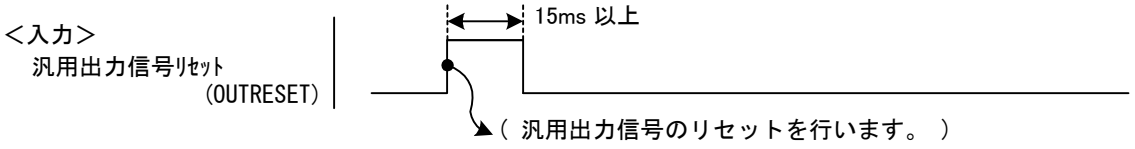
(16) SAFEPOS (待避点復帰入力 /: 待避点復帰中出力)



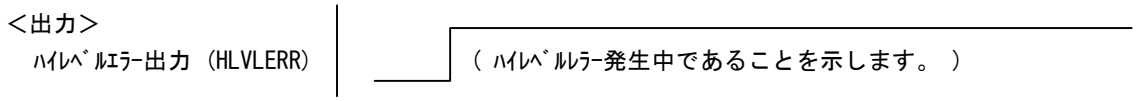
(17) BATERR(バッテリ電圧低下出力)



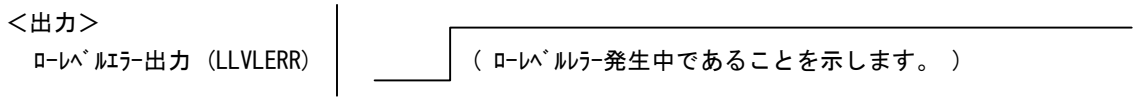
(18) OUTRESET (汎用出力信号リセット要求入力)



(19) HLVLERR (ハイレベルエラー発生中出力)



(20) LLVLERR(ローレベルエラー発生中出力)



(21) CLVLERR(警告レベルエラー発生中出力)

<出力>
警告ﾚﾍﾞﾙｴﾗｰ出力
(CLVLERR)

(警告ﾚﾍﾞﾙｴﾗｰ発生中であることを示します。)

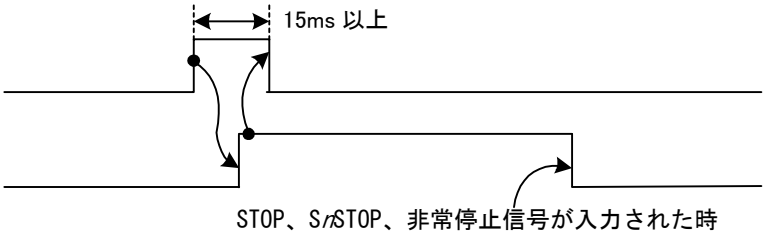
(22) EMGERR(非常停止中出力)

<出力>
非常停止出力 (EMGERR)

(非常停止が発生中であることを示します。)

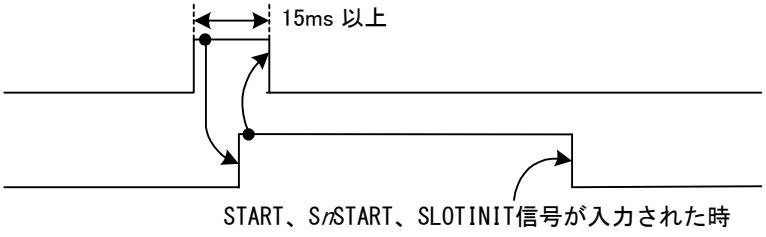
(23) SnSTART (スロット n 始動入力 / スロット n 運転中出力)

<入力>
スロット *n* 始動入力
(S*n*START)
<出力>
スロット n 運転中出力
(SnSTART)



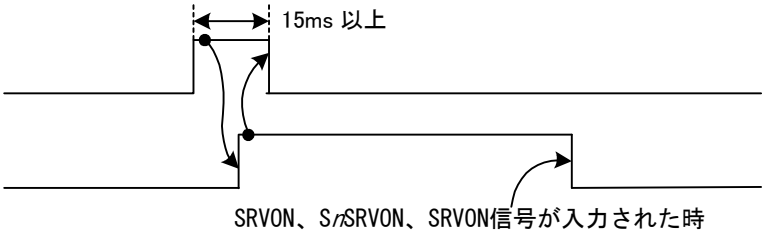
(24) SnSTOP (スロット n 停止入力 / スロット n 中断中出力)

<入力>
スロット *n* 停止入力
(S*n*STOP)
<出力>
スロット n 中断中出力
(S*n*STOP)



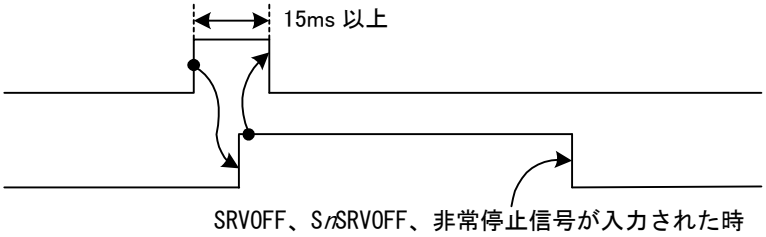
(25) MnSRVOFF (メカ n サーボ OFF 入力 / メカ n サーボ ON 不可出力)

<入力>
メカ *n* サーボ OFF 入力
(M*n*SRVOFF)
<出力>
メカ *n* サーボON 不可出力
(M*n*SRVOFF)

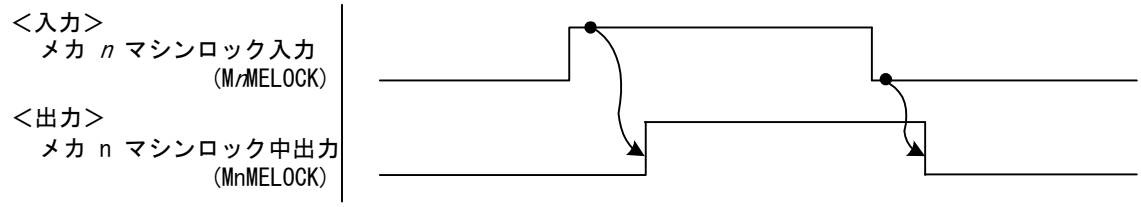


(26) MnSRVON (メカ n サーボ ON 入力 / メカ n サーボ ON 中出力)

<入力>
メカ *n* サーボ ON 入力
(M*n*SRVON)
<出力>
メカ *n* サーボON 中 出力
(M*n*SRVON)

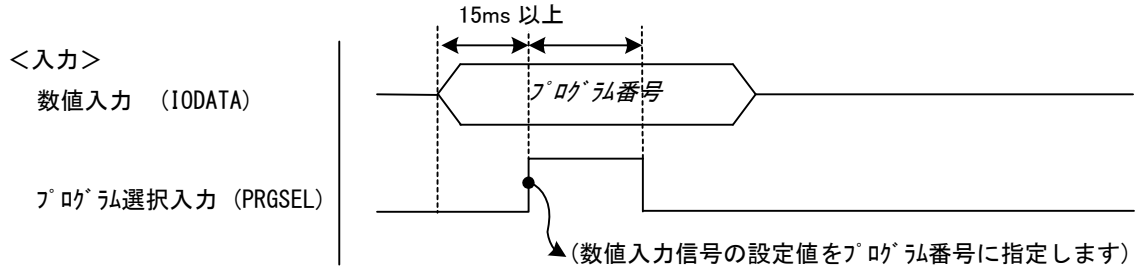


(27) MnMELOCK (メカ n マシンロック入力 / メカ n マシンロック中出力)



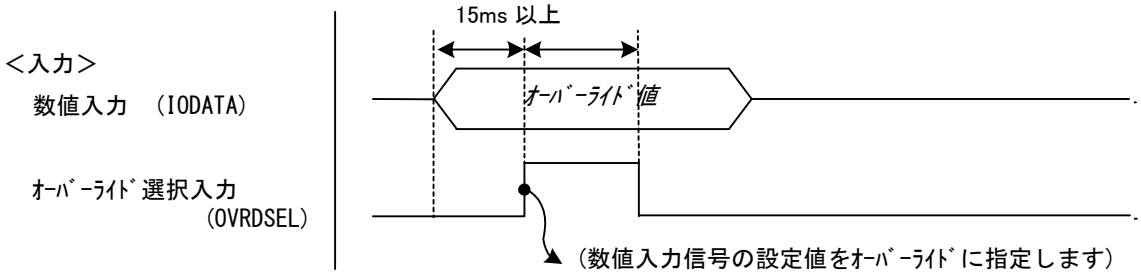
(28) PRGSEL(プログラム選択入力)

※ 数値入力(IODATA)と合わせて使用します。



(29) OVRDSEL (オーバーライド選択入力)

※ 数値入力(IODATA)と合わせて使用します。

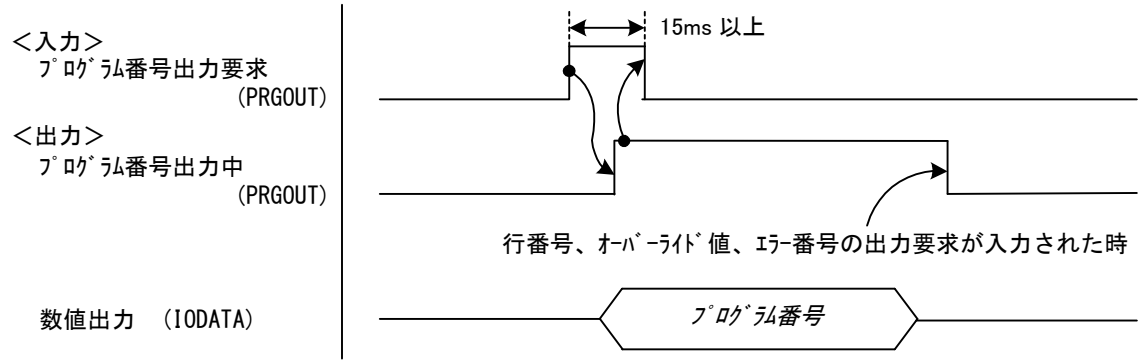


(30) IODATA (数値入力 / 数値出力)

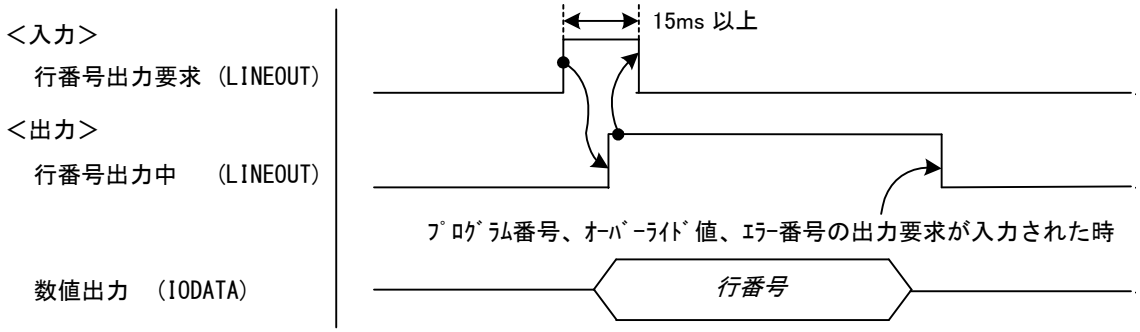
※ PRGSEL、OVRDSEL、PRGOUT、LINEOUT、OVRDOUT、ERROUT と合わせて使用します。

(31) PRGOUT (プログラム番号出力要求入力 / プログラム番号出力中)

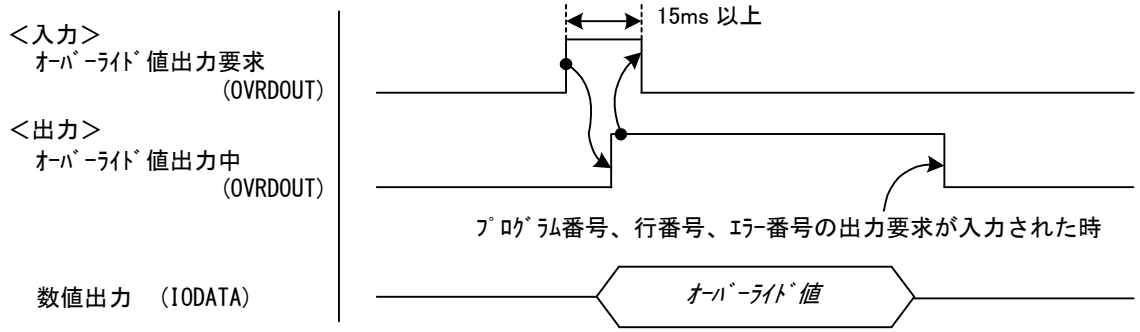
※ 数値出力(IODATA)と合わせて使用します。



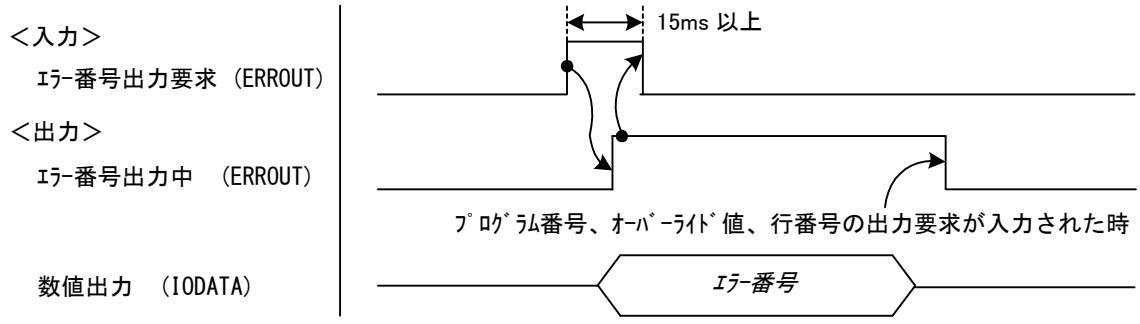
(32) LINEOUT (行番号出力要求入力 / 行番号出力中)
※ 数値出力(IODATA)と合わせて使用します。



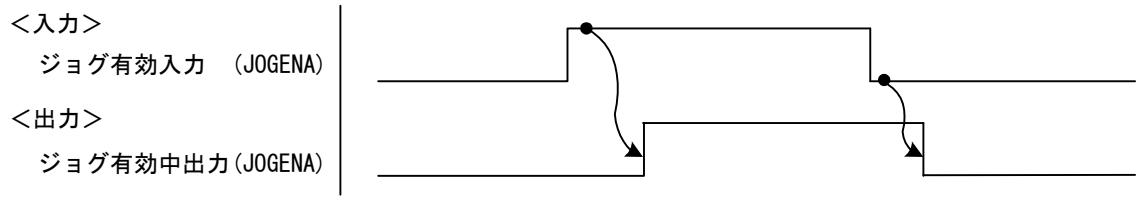
(33) OVRDOUT (オーバーライド値出力要求 / オーバーライド値出力中)
※ 数値出力(IODATA)と合わせて使用します。



(34) ERROUT (エラー番号出力要求 / エラー番号出力中)
※ 数値入力(IODATA)と合わせて使用します。

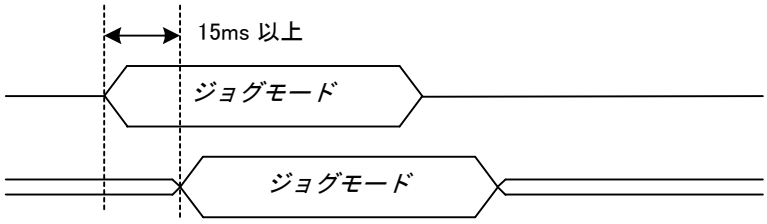


(35) JOGENA (ジョグ有効入力 / ジョグ有効中出力)



(36) JOGM(ジョグモード入力 / ジョグモード出力)

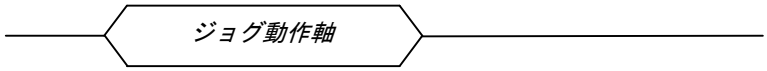
＜入力＞
ジョグモード入力 (JOGM)
＜出力＞
ジョグモード出力 (JOGM)



(ｼﾞｮｸﾞﾓｰﾄﾞ入力信号での設定値をｼﾞｮｸﾞﾓｰﾄﾞ出力にて返信します。)

(37) JOG+(ジョグ送りプラス側 8 軸分入力)

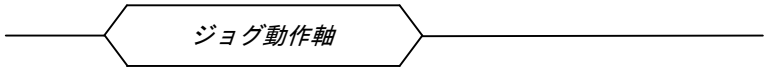
＜入力＞
ジョグ送りﾌﾟﾗｽ側 8 軸分
(JOG+)



(ﾌﾟﾗｽ方向のジョグ動作を行う軸を指定します。)

(38) JOG-(ジョグ送りマイナス側 8 軸分入力)

＜入力＞
ジョグ送りﾏｲﾅｽ側 8 軸分
(JOG-)



(ﾏｲﾅｽ方向のジョグ動作を行う軸を指定します。)

(39) HNDCNTLn (メカ n ハンド出力信号状態)

＜出力＞
メカ *n* ﾊﾝﾄﾞ出力信号状態
(HNDCNTL*n*)

(ﾊﾝﾄﾞの出力信号状態を示します。)

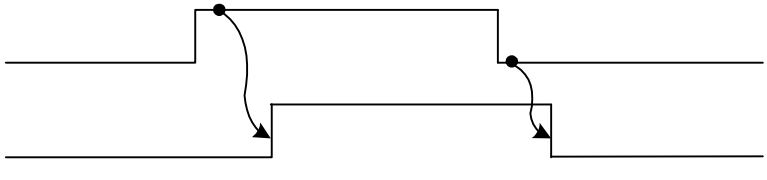
(40) HNDSTSn(メカ n ハンド入力信号状態)

＜出力＞
メカ *n* ﾊﾝﾄﾞ入力信号状態
(HNDSTS*n*)

(ﾊﾝﾄﾞの入力信号状態を示します。)

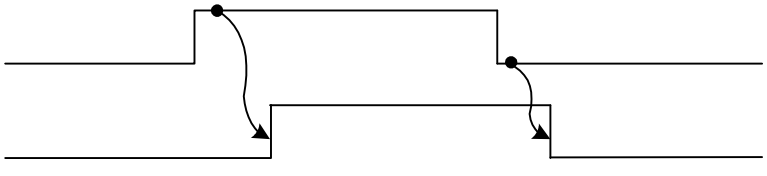
(41) HNDERRn(メカ n ハンドエラー入力信号 / メカ n ハンドエラー中出力)

＜入力＞
メカ *n* ﾊﾝﾄﾞｴﾗｰ入力
(HNDERR*n*)
＜出力＞
メカ n ﾊﾝﾄﾞｴﾗｰ中出力
(HNDERRn)



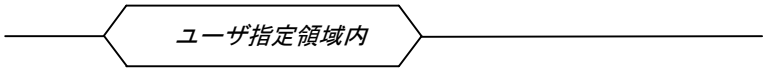
(42) AIRERRn （ メカ n 空気圧エラー入力信号 / メカ n 空気圧エラー出力中 ）

＜入力＞
メカ *n* 空気圧ｴﾗｰ入力
(AIRERR*n*)
＜出力＞
メカ n 空気圧ｴﾗｰ中出力
(AIRERRn)



(43) USRAREA(ユーザ定義領域 8 点出力)

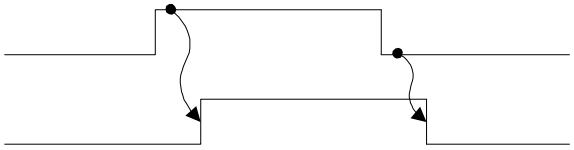
＜出力＞
ユーザ指定領域８点
(USRAREA)



(領域１～８で設定した領域内にいることを示します。)

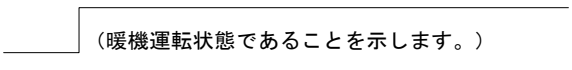
(44) MnWUPENA(メカ n 暖機運転モード有効入力 / メカ n 暖機運転モード出力)

〈入力〉
メカ*n*暖機運転モード有効入力
(M*n*WUPENA)
〈出力〉
メカ*n*暖機運転モード出力
(M*n*WUPENA)



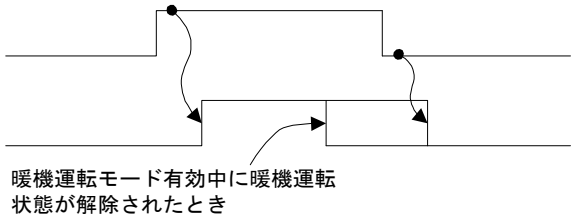
(45) MnWUPMD(メカ n 暖機運転状態出力)

〈出力〉
メカ*n*暖機運転状態出力
(M*n*WUPMD)



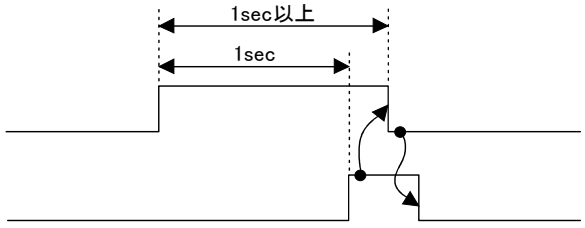
※ メカ n 暖機運転モード有効入力 (MnWUPENA) と同時に割り付けたときは次のようになります。

〈入力〉
メカ*n*暖機運転モード有効入力
(M*n*WUPENA)
〈出力〉
メカ*n*暖機運転状態出力
(M*n*WUPMD)

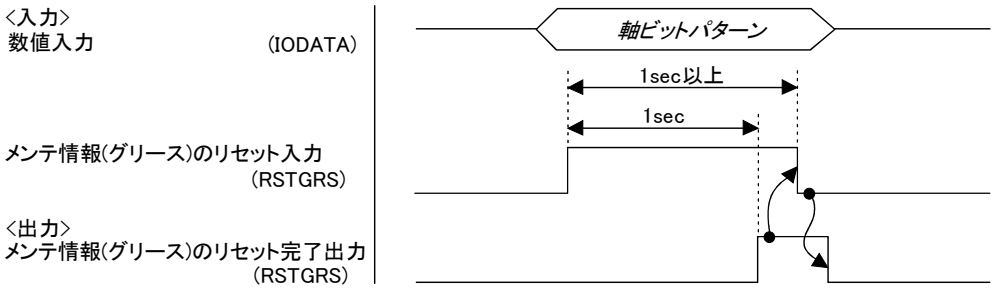


(46) RSTBAT(バッテリ積算時間のリセット)

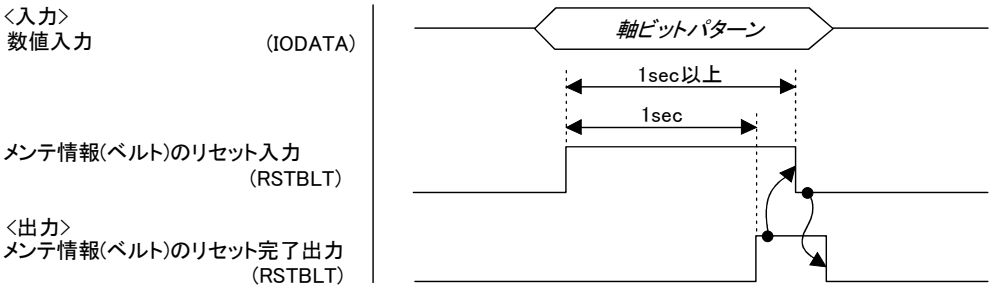
〈入力〉
バッテリ積算時間のリセット入力
(RSTBAT)
〈出力〉
バッテリ積算時間のリセット完了出力
(RSTBAT)



(47) RSTGRS(メンテ予報のリセット (グリース))



(48) RSTBAT(メンテ予報のリセット (ベルト))



6.5.2 タイミングチャート例

(1) 外部信号操作タイムチャート (その 1)

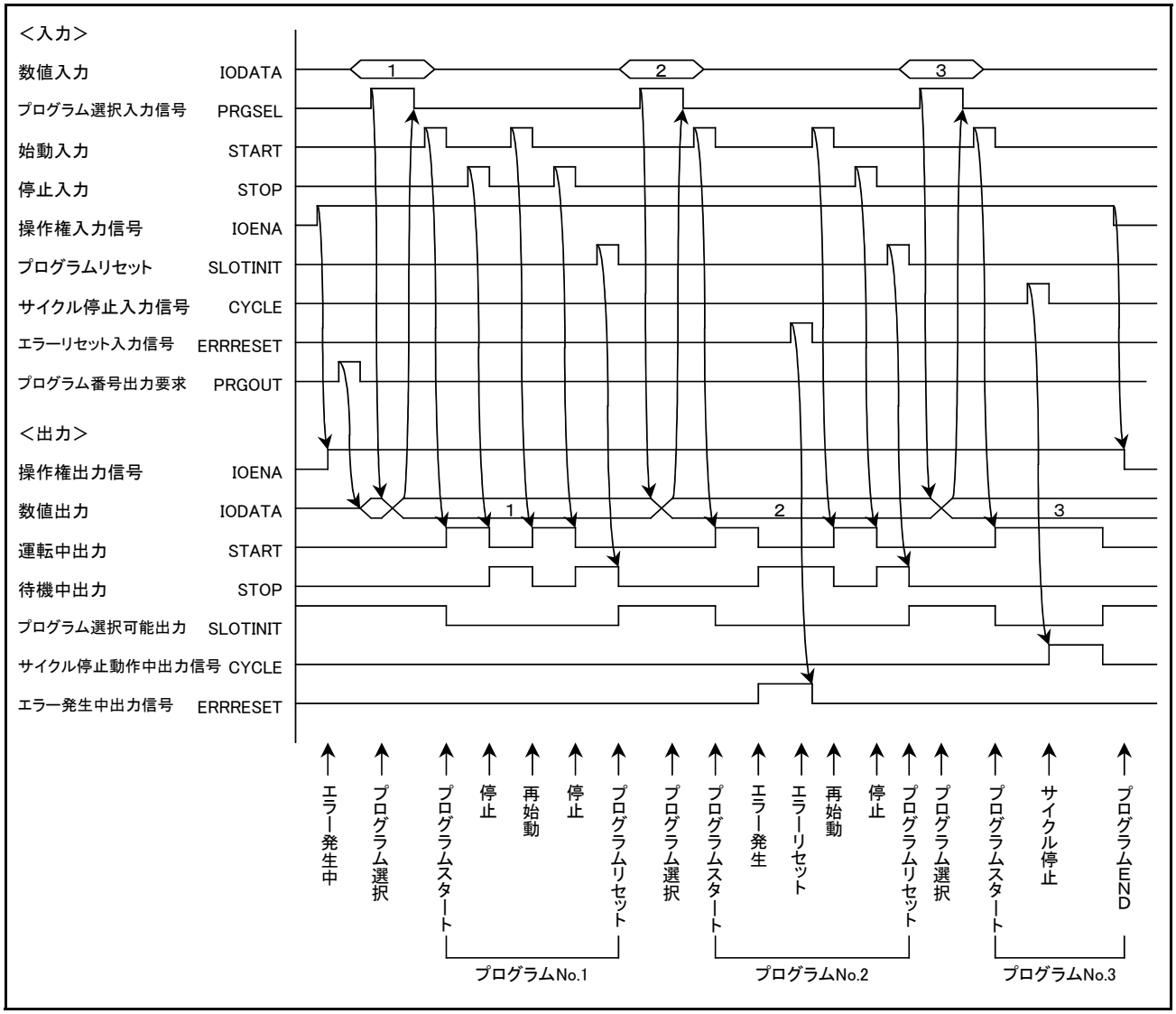


図 6-4 : 外部操作タイムチャート例 (その 1)

(2) 外部信号操作タイムチャート (その 2)

図 6-5 に外部信号によるサーボオン / オフ、プログラム選択、オーバーライド選択、始動、行番号出力などのタイムチャート例を示します。

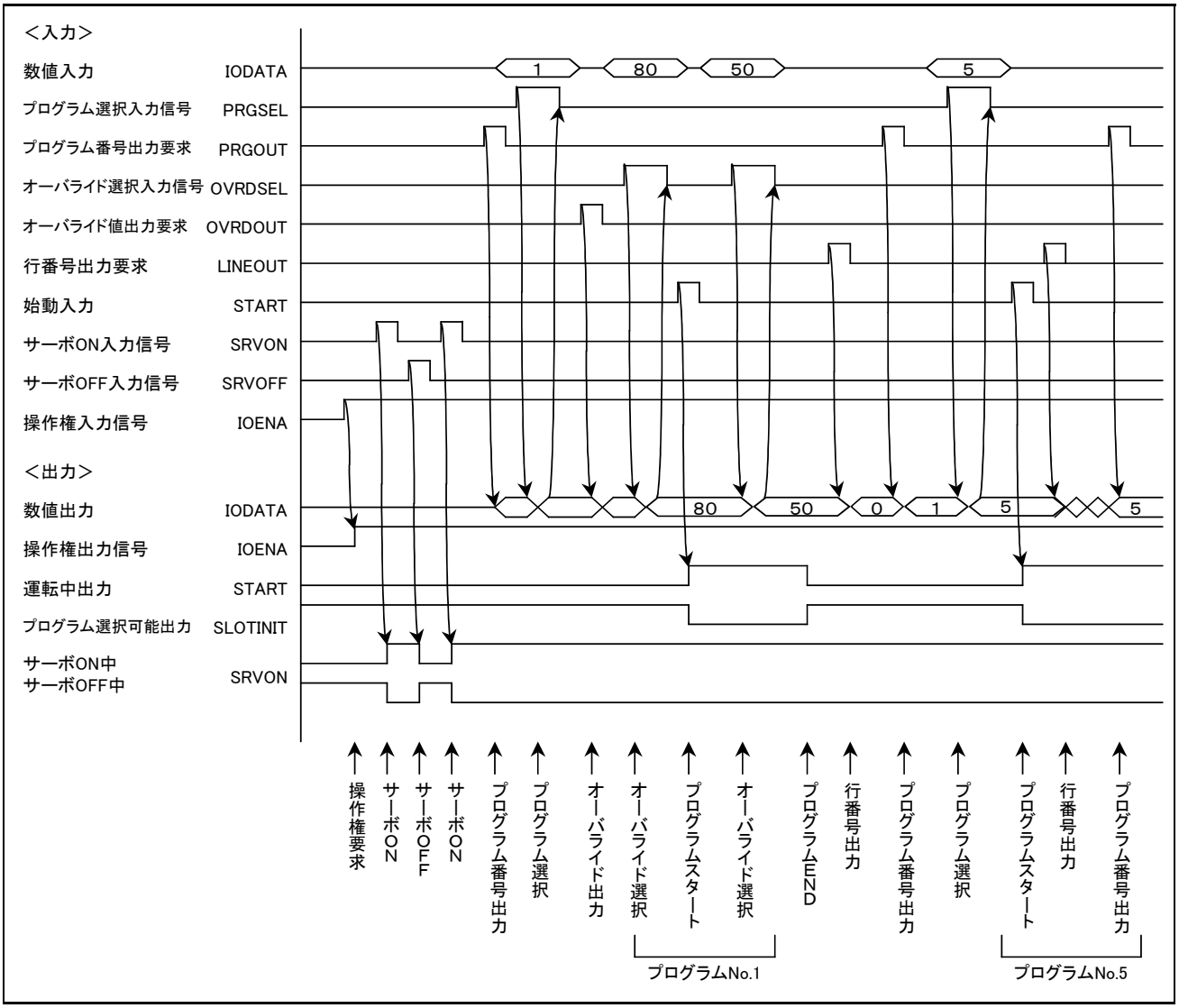


図 6-5 : 外部信号操作タイムチャート (その 2)

(3) 外部信号操作タイムチャート (その 3)

図 6-6 に外部信号によるエラーリセット、汎用出力リセット、プログラムリセットなどのタイムチャート例を示します。

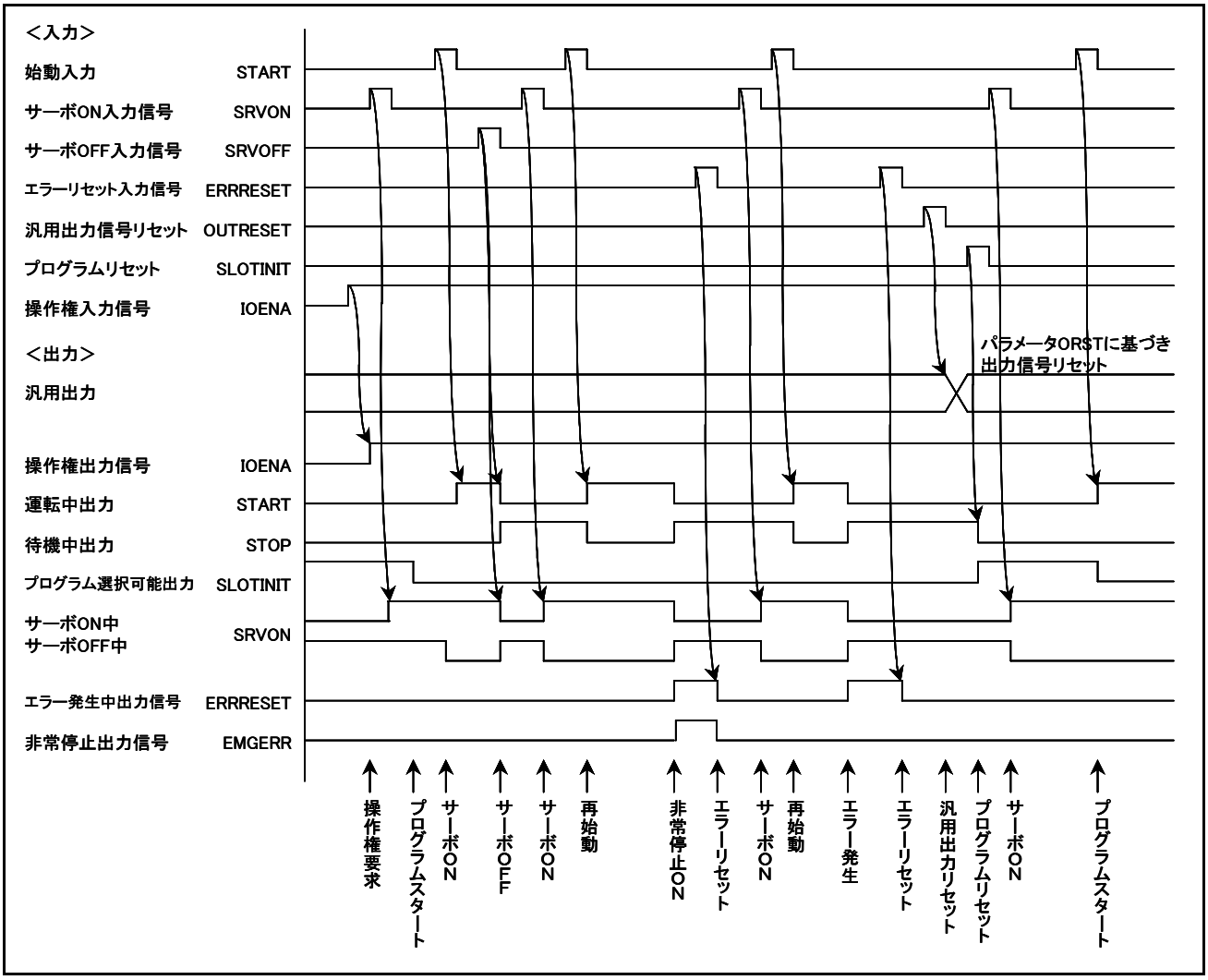


図 6-6 : 外部信号操作タイムチャート (その 3)

(4) 外部信号操作タイムチャート (その 4)

図 6-7 に外部信号によるジョグ操作、待避点復帰、プログラムリセットなどのタイムチャート例を示します。

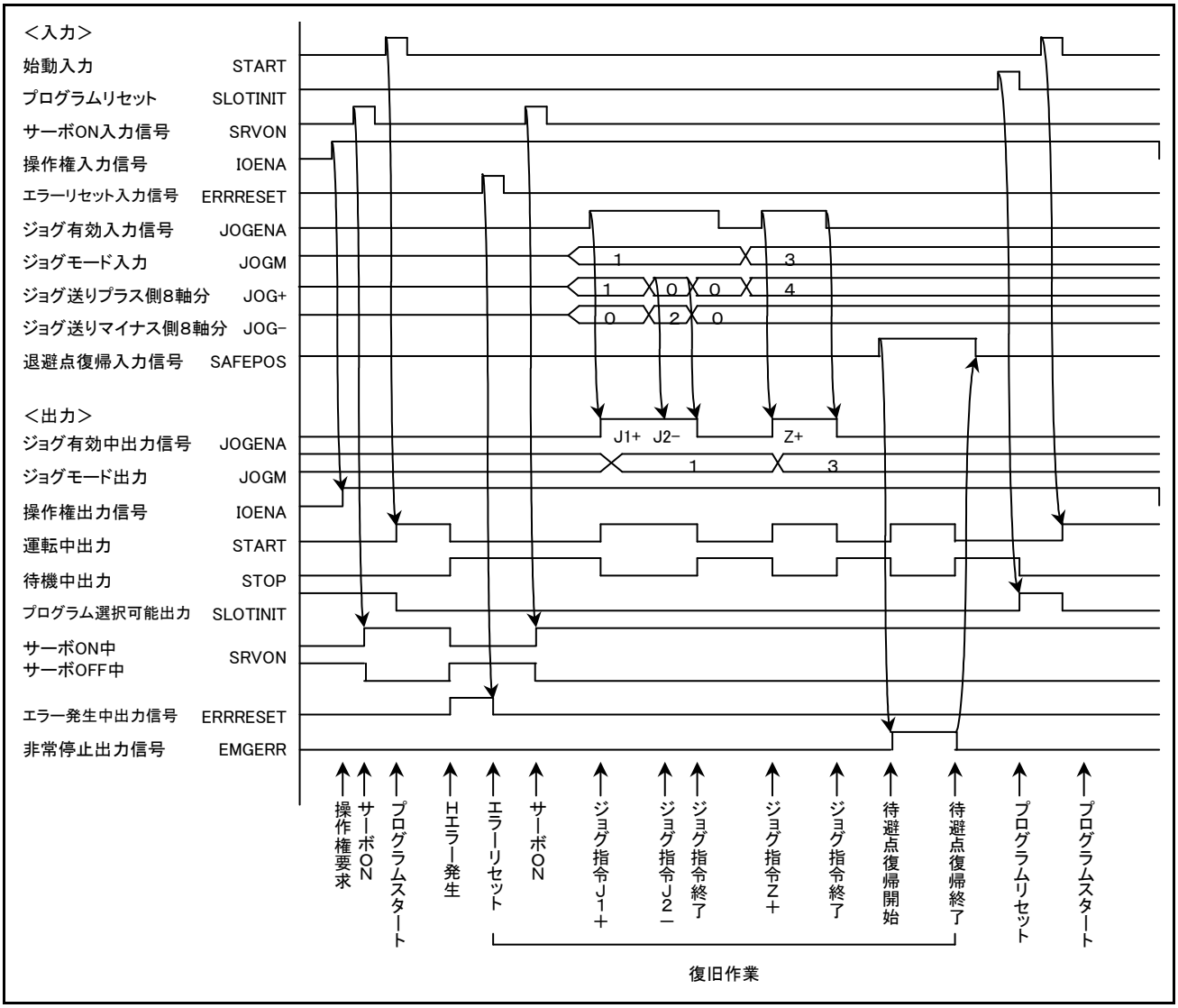


図 6-7 : 外部信号操作タイムチャート (その 4)

(5) 外部信号操作タイムチャート (その 5)

図 6-8 に指定した位置データを出力する場合のタイムチャート例を示します。

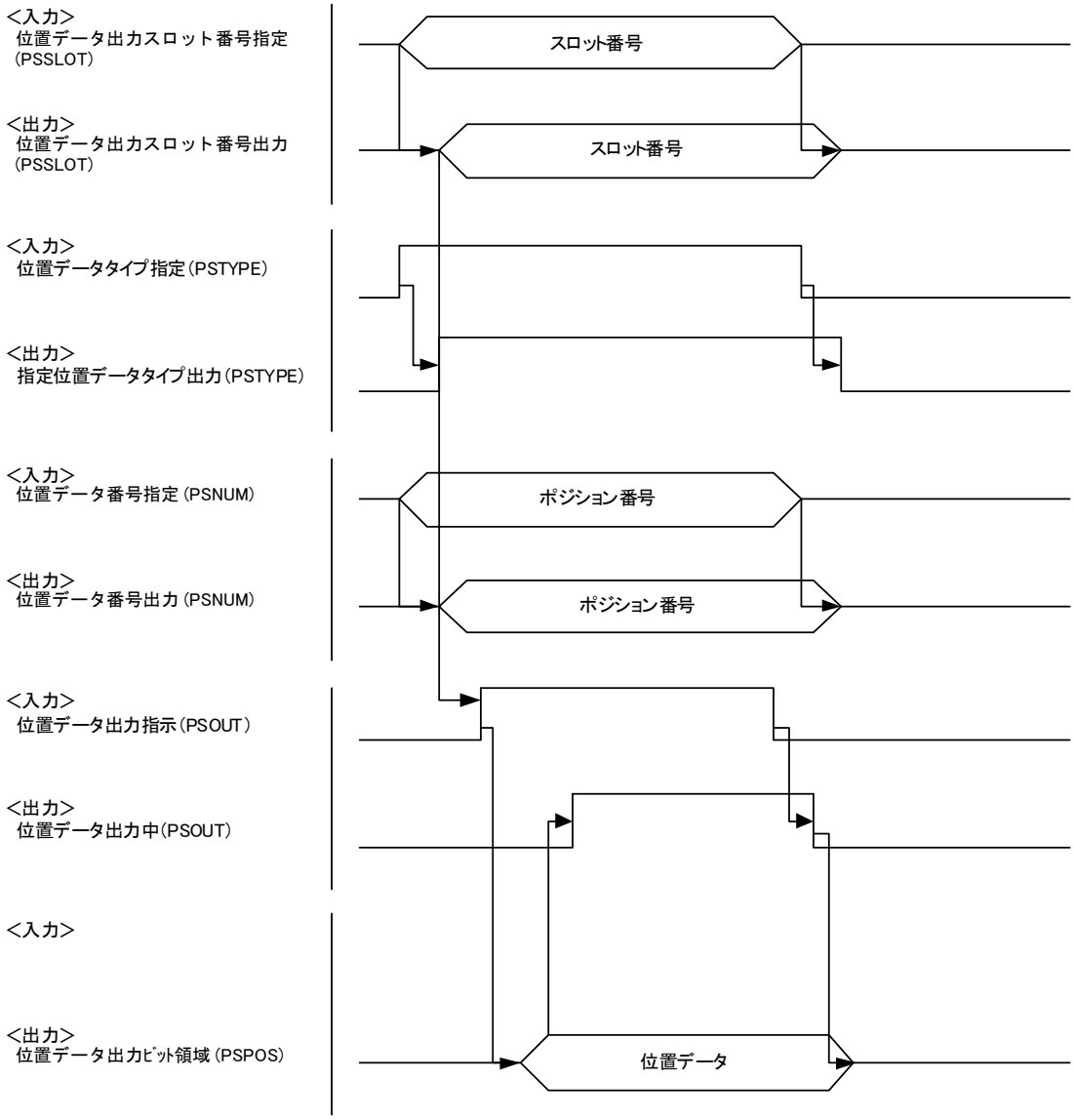


図 6-8 : 外部信号操作タイムチャート (その 5)

[注意]

(1) パラメータ「PSPOS」に指定した信号番号から 320 点分の出力信号領域が存在しない場合は、エラー : 7081(パラメータの値が範囲外なので書き込めません) が発生し、位置データのモニタを実行することができません。必ず 320 点分の連続領域を確保してください。

(2) パラメータ「PSSLOT」に設定した信号番号の範囲が 6bit より大きい場合は、エラー : 7081(パラメータの値が範囲外なので書き込めません) が発生します。

(3) パラメータ「PSNUM」に設定した位置データ番号の範囲が 16bit より大きい場合は、エラー : 7081(パラメータの値が範囲外なので書き込めません) が発生します。

(4) 位置データ出力指示 (PSOUT) 入力中は、スロット番号指定、位置データタイプ指定、位置データ番号指定が変更されても、その指定は無効となります。この場合、位置データ出力指示 (PSOUT) の入力を一旦 OFF してから再度 ON してください。どの位置データが出力されているかは、スロット番号出力 (PSSLOT)、位置データタイプ出力 (PSTYPE)、位置データ番号出力 (PSNUM) をご確認ください。

(5) 指定されたスロットにプログラムがロードされていない場合は、各軸の値を "&H7FFFFFFF" で出力します。

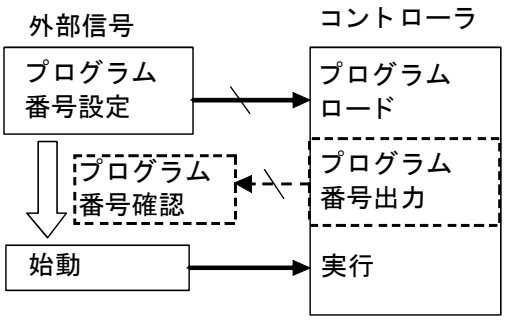
(6) 指定された位置データが存在しない場合は、各軸の値を "&H80000000" で出力します。

(7) 位置データ出力中に、指定スロットで実行中のプログラムが切り替わった場合 (CallP コマンドや XRun コマンド、パラメータ「PRGSEL」) は、各軸の値を "&H80000000" で出力します。

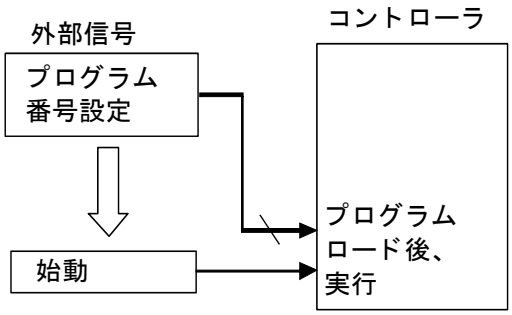
6.6 外部信号からのプログラム選択･実行方法

6.6.1 種類

外部信号からプログラムを選択し実行する方法として、以下の 2 通りがあります。
(a) 指定のプログラムがロードされたことを確認してから実行を開始する方法
(b) 始動信号と同時にプログラム番号を指定して実行を開始する方法



(a)：指定プログラムのロードを確認



(b)：始動信号と同時にプログラムを指定

6.6.2 実行方法の選択

表 6-8 のパラメータを設定して外部信号によるプログラムの実行方法 (上記 (a) または (b)) を指定します。

表 6-8：プログラム選択方法切替えパラメータ

ﾊﾟﾗﾒｰﾀ名	機能概要
PST	ﾌﾟﾛｸﾞﾗﾑ選択方法を切替える。 0：指定のプログラムがロードされたことを確認してから実行を開始する 1：始動信号と同時にプログラム番号を指定して実行を開始する

6.6.3 関連する入出力パラメータ

表 6-9：入出力信号パラメータ

ﾊﾟﾗﾒｰﾀ名	区分	機能概要
IOENA	入力	外部信号制御の操作を有効 / 無効を切替える
PRGOUT	入力	ﾀｽｸｽﾛｯﾄ 1 のﾌﾟﾛｸﾞﾗﾑ番号を数値出力 (IODATA) に出力させます。
IODATA	入力 / 出力	数値入力 / 数値出力を設定。設定された範囲のﾋﾞｯﾄ状態をﾊﾞｲﾅﾘｰ値として読み取れるよう設定する。
PRGSEL	入力	数値入力信号での設定値をプログラム番号に指定します。
START	入力	プログラムの始動を行う。

6.6.4 操作手順

ここでは、例として入出力信号を表 6-10 に示した割付けとします。

表 6-10：入出力信号割付

ﾊﾟﾗﾒｰﾀ名	入力ビット	出力ビット
IOENA	5	3
PRGOUT	7	−
IODATA	8 ～ 11	8 ～ 11
PRGSEL	6	−
START	3	−

(1) 指定のプログラムがロードされたことを確認してから実行を開始する場合

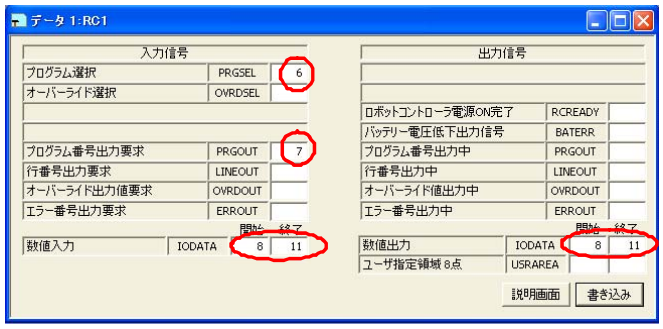
1) パラメータ「PST」の設定値を 0 にします。(初期設定値)

2) 表 6-10 の入出力ビットをパラメータに割り付けます。

下図は ToolBox2 の【パラメータ設定】→【専用入出力割り当て】の設定画面です。図中の丸印の箇所を設定します。

【専用入出力割り当て】:【汎用 1】画面

【専用入出力割り当て】:【データ】画面



3) プログラム選択・始動の手順

①入力信号 IOENA(ﾋﾞｯﾄ 5) を ON し、操作権を取得します。

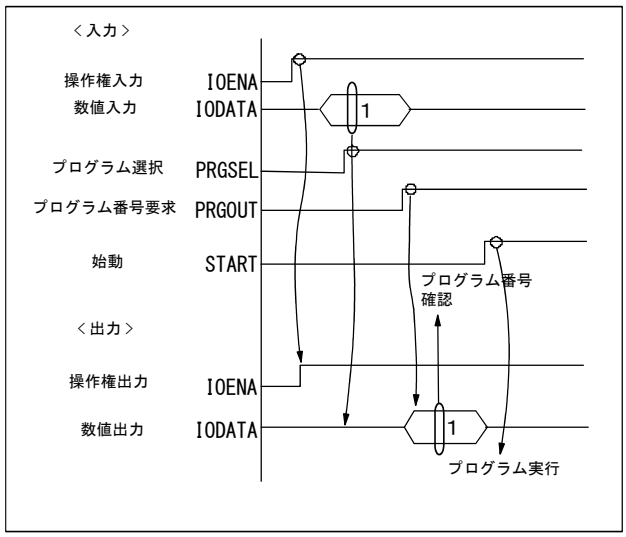
②出力信号 IOENA(ﾋﾞｯﾄ 3) を確認し、ON であれば、操作権が取得できたことになります。

③プログラム番号を入力用 IODATA(ﾋﾞｯﾄ 8 ～ 11) にバイナリ値で設定します。

④ PRGSEL(ﾋﾞｯﾄ 6) を ON します。
入力用 IODATA(ﾋﾞｯﾄ 8 ～ 11) で設定した番号のプログラムが読み込まれます。

⑤ PRGOUT(ﾋﾞｯﾄ 7) を ON します。
出力用 IODATA(ﾋﾞｯﾄ 8 ～ 11) に、読み込まれたプログラム番号が出力されます。番号が一致しているか確認できます。

⑥ START(ﾋﾞｯﾄ 3) を入力し、プログラムを実行させます。



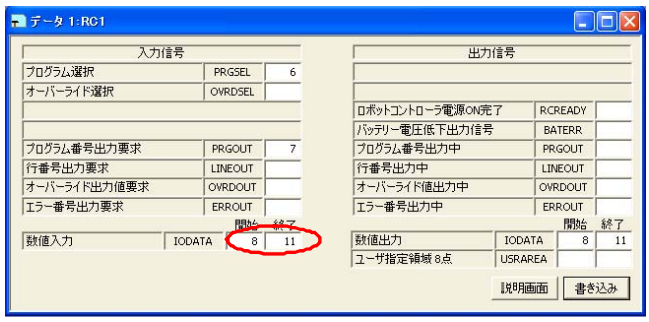
(2) 始動信号と同時にプログラム番号を指定して実行を開始する場合

1) パラメータ「PST」の設定値を 1 にします。(初期値は 0)

2) 表 6-10 の入出力ビットをパラメータに割り付けます。
下図は ToolBox2 の【パラメータ設定】→【専用入出力割り当て】の設定画面です。図中の丸印の箇所を設定します。

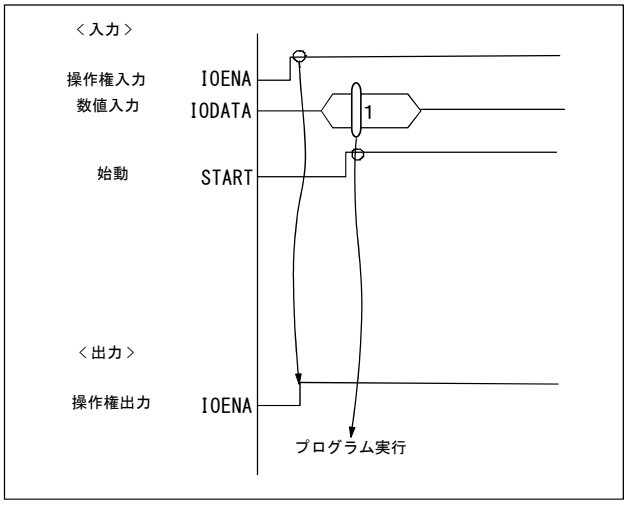
【専用入出力割り当て】:【汎用 1】画面

【専用入出力割り当て】:【データ】画面



3) プログラム選択・始動の手順

①入力信号 IOENA(ﾋﾞｯﾄ 5) を ON し、操作権を取得します。
②出力信号 IOENA(ﾋﾞｯﾄ 3) を確認し、ON であれば、操作権が取得できたことになります。
③プログラム番号を入力用 IODATA(ﾋﾞｯﾄ 8 ～ 11) にバイナリ値で設定します。
④ START(ﾋﾞｯﾄ 3) を入力し、プログラムを実行させます。



注 1) パラメータを変更した場合は、コントローラの電源を入れ直してしてください。

注 2) プログラム実行時、サーボ ON、プログラム初期化などは省略しています。本書の 516 ページの「6.3 専用入出力」～ 526 ページの「6.4 信号の有効無効状態」を参照ください。

6.7 非常停止入力

非常停止入力の配線等に関しては別冊の「コントローラセットアップと基本操作から保守まで」編を参照してください。

6.7.1 非常停止入力時のロボットの挙動

ロボット動作中に非常停止信号が入力されますとサーボ電源をハード的に遮断します。非常停止入力後のロボット先端の軌跡と停止位置は特定できません。ロボットの速度、ツールの負荷状況によってはオーバーランすることがあります。

7 付録

7.1 構造フラグについて

構造フラグとはロボットの姿勢を表すフラグのことです。
ロボットは、X，Y，Z，A，B，C による位置データによりロボットのハンド先端を記憶します。しかし、同じ位置データでもロボットのとりうる姿勢は複数あります。この姿勢を表すものが構造フラグで、位置定数(X，Y，Z，A，B，C)(FL1、FL2)の内の FL1 でその姿勢を記憶しています。
以下に、構造フラグの種類を示します。

■垂直多関節型ロボットの場合

(1) RIGHT/LEFT

J1 軸の回転中心鉛直上に対するフランジ中心の位置(R)を表します。(5 軸タイプ)
J1 軸の回転中心鉛直上に対する J5 軸の回転中心の位置(P)を表します。(6 軸タイプ)

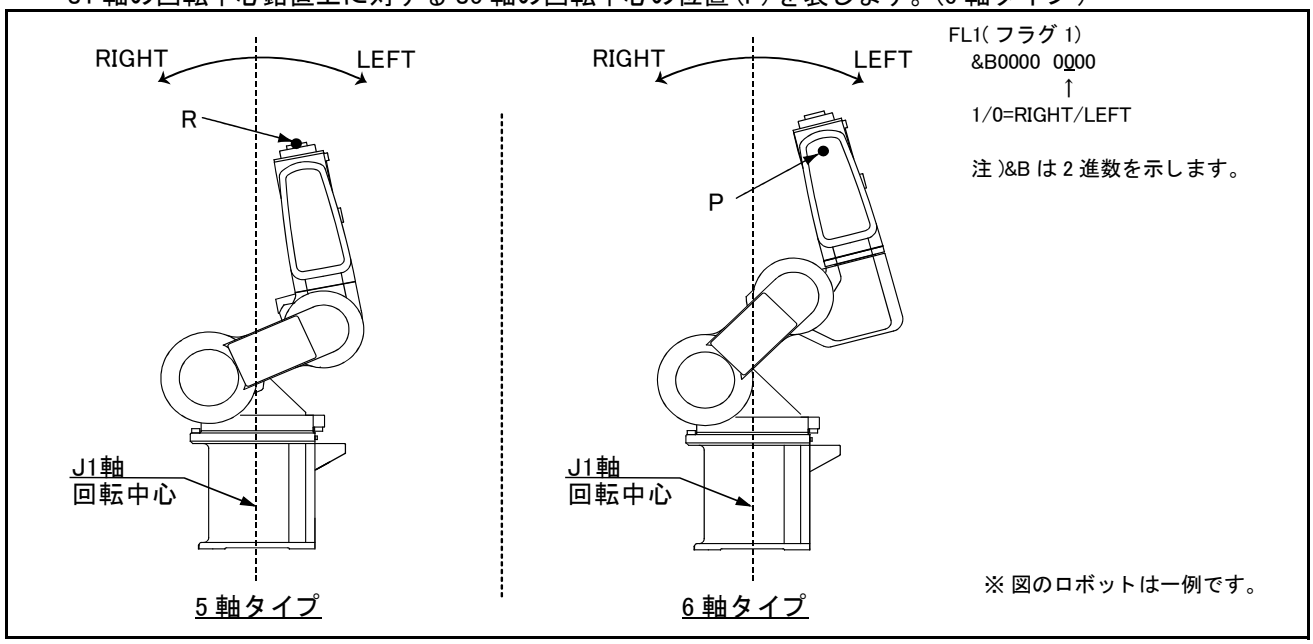


図 7-1：構造フラグ (RIGHT/LEFT)

(2) ABOVE/BELOW

J2 軸の回転中心から J3 軸の回転中心方向に対する J5 軸の回転中心の位置(P)を表します。

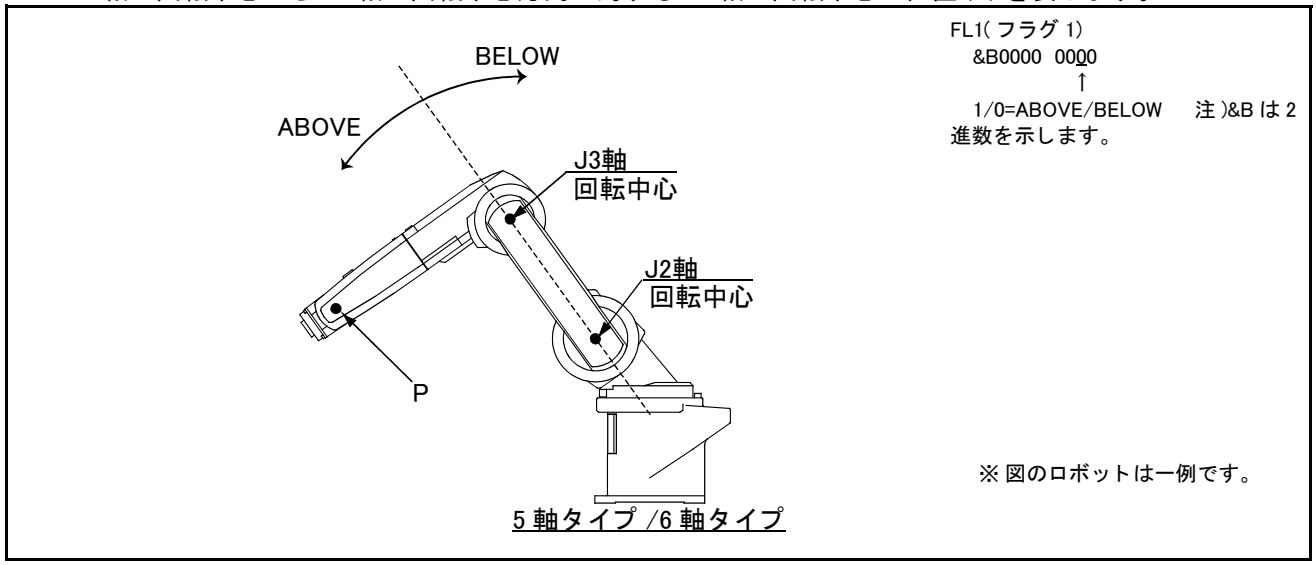
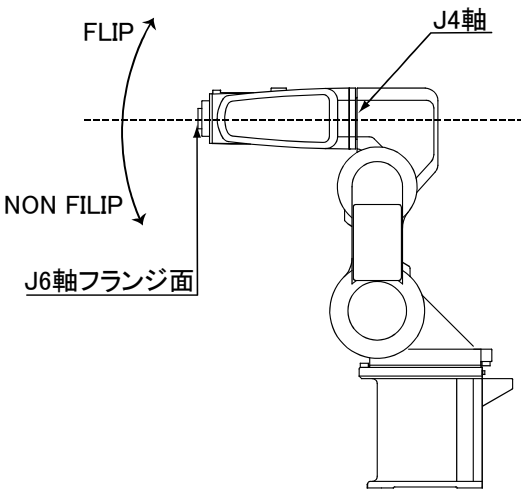


図 7-2：構造フラグ (ABOVE/BELOW)

(3) NONFLIP/FLIP(6 軸タイプのみ)

J4 軸の回転中心から J5 軸の回転中心方向に対するフランジ面の向きを表します。



FL1(フラグ 1)
&B0000 0000
↑
1/0=NONFLIP/FLIP

注)&B は 2 進数を示します。

※ 図のロボットは一例です。

図 7-3 : 構造フラグ (NONFLIP/FLIP)

■水平多関節型ロボットの場合

(1) RIGHT/LEFT

J1 軸の回転中心から J2 軸の回転中心を通る直線上に対する先端軸の位置を表します。

FL1(フラグ 1)
&B0000 0000
↑
1/0=RIGHT/LEFT

注)&B は 2 進数を示します。

※ 図に示すロボットは一例です。

図 7-4 : 構造フラグ (RIGHT/LEFT)

7.2 多機能電動ハンドの使用方法

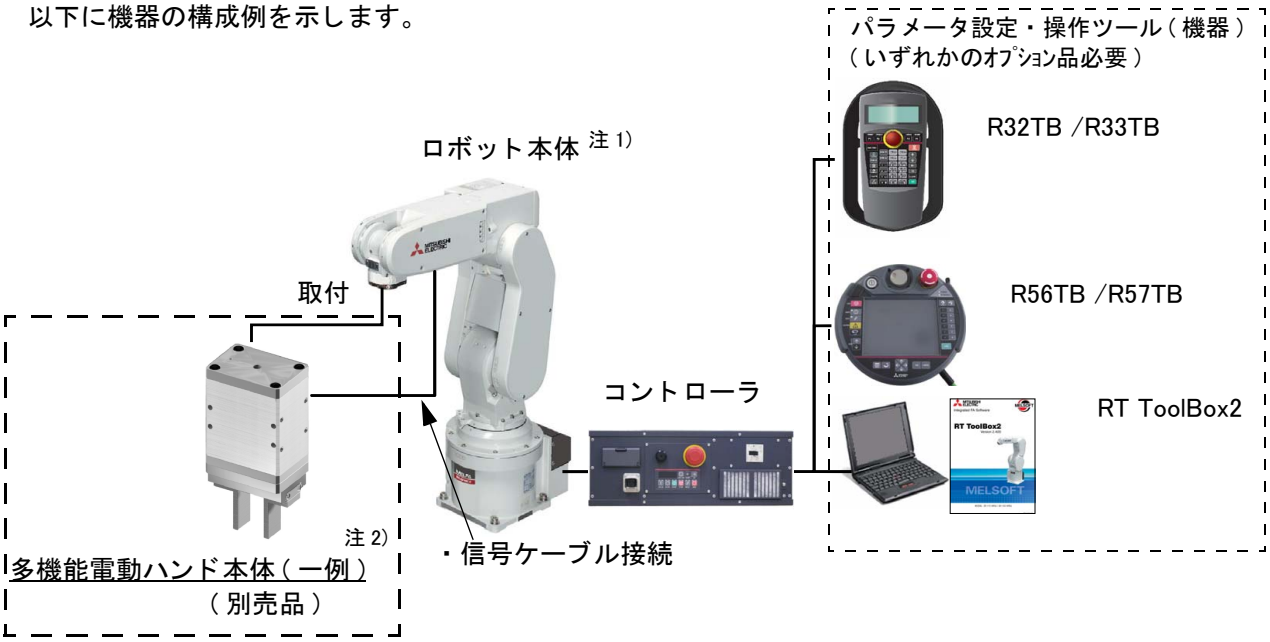
TAIYO 社製多機能電動ハンドをご使用になる際の設定、及び操作方法について説明します。本書に記載なき事項は多機能電動ハンド製品に付属の取扱説明書をご参照願います。

注）以下、本章での〝電動ハンド〟とは〝多機能電動ハンド〟のことを示します。

7.2.1 概要

(1) 機器構成（例）

以下に機器の構成例を示します。



注 1) 図は RV-2F の構成例です。
RV-2F では標準仕様のロボットで多機能電動ハンドのご使用が可能です。
他のロボットでは特殊仕様となります。（垂直多関節 6 軸タイプのみ）別途発行の納入仕様書を参照願います。

注 2) 多機能電動ハンド本体の図は一例を記載しています。多機能電動ハンドを構成する付属品、および仕様等につきましては多機能電動ハンドに付属の取扱説明書を参照願います。

⚠ 注意

調整作業などで、多機能電動ハンド部分を調整する時は、必ず駆動電源を OFF してください。そうでない場合は、手や指を挟まれる危険性があります。
また、ロボット本体のサーボ OFF 操作でも多機能電動ハンドは駆動電源 ON 状態を保持します。これは、把持しているワークを不用意に落下させないための処置です。
教示や調整作業中には多機能電動ハンドの駆動電源 ON/OFF 状態を把握の上、安全に作業をおこなってください。
T/B からの操作で、多機能電動ハンドの駆動電源を OFF することができます。
ハンド開 / 閉等の操作や動作させる命令の実行で、自動的に駆動電源を ON します。

図 7-5：多機能電動ハンド使用時の構成例

7.2.2 仕様

表 7-1：仕様

項目	単位	仕様	備考
最大接続台数	台	1	ハンド 1、ハンド 2、ハンド 3 と示します。
接続仕様	－	リモート I/O 接続	
チャネル番号	－	ﾁｬﾈﾙ 1	
局番（入出力番号）	－	ハンド 1：局番 5(700 ～ 731) ハンド 2：局番 6(732 ～ 763) ハンド 3：局番 7(764 ～ 795)	局番は多機能電動ハンド本体のスイッチで設定します。 局番の設定で、3 種類のハンドを区別できます。同時に 1 台のハンドをご使用いただけます。 ※ 左記入出力番号の信号は多機能電動ハンド用に占有します。ユーザプログラムなどでの信号出力はできません。
教示点数（最大）	点	32	ひとつのハンドあたりの最大教示点数です。

7.2.3 制限事項

(1) 使用可能な機器とソフトウェアバージョン

多機能電動ハンドは表 7-2 に示すロボットと機器、ツール、および対象のソフトウェアバージョンでご使用できます。

表 7-2：使用可能な機器とソフトウェアバージョン

機器	コントローラ	R32TB/R33TB	R56TB/R57TB	RT ToolBox2
使用可能なソフトウェアバージョン	R3 以降	Ver. 1.6 以降	Ver. 2.5 以降	Ver. 2.1 以降

【注意】：多機能電動ハンドは、電源 ON の後、動作を始める前に一度は原点復帰が必要です。
このためコンティニュー機能（パラメータ：CTN で指定）には対応していません。

(2) 局番設定

局番は、多機能電動ハンドのロータリースイッチで、5、6、7 のいずれかに設定してご使用願います。

以下にハンド番号と局番の関係を示します。ハンド番号は、T/B などでの操作の対象とするハンド番号の切替や、設定するパラメータ名称とロボット言語や、状態変数で指定するハンド番号、およびエラー番号に関連付けられます。

表 7-3：ハンド番号と局番の関係

名称	ハンド番号	局番
ハンド 1	1	5
ハンド 2	2	6
ハンド 3	3	7

注）同時に 1 台のハンドをご使用いただけます。

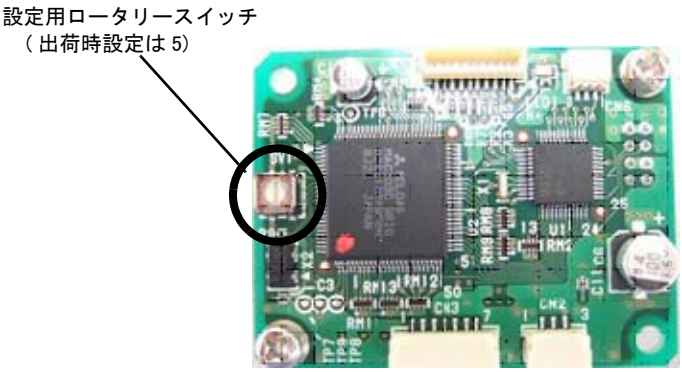


図 7-6：局番設定

(3) 多機能電動ハンドの駆動電源 (ロボット本体との関係)

多機能電動ハンドの駆動電源 ON/OFF 状態は、ロボット本体のサーボ ON/OFF と同期していません。ロボット本体を非常停止にしても多機能電動ハンドの駆動電源は OFF されません。また、ロボット本体のサーボ OFF 操作でも、多機能電動ハンドの駆動電源は ON 状態を保持します。
多機能電動ハンドの駆動電源 OFF は、T/B の操作でおこないます。また、駆動電源 ON は、動作させる操作や命令の実行で自動的に ON します。

表 7-4 : 多機能電動ハンドの駆動電源 ON/OFF

操作	駆動電源 ON/OFF 状態	備考
T/B からのハンドサーボ OFF	OFF	
ロボットの非常停止 ON	その時の状態を維持	その時の状態を維持するため、不用意に把持したワークの落下を防止します。
ロボット本体のサーボ OFF	その時の状態を維持	
ロボット本体のサーボ ON	その時の状態を維持	
T/B からのハンド開閉操作	ON	自動で駆動電源を ON します。
ハンド動作命令の実行	ON	

7.2.4 操作上の注意事項

(1) 原点復帰

多機能電動ハンドは、電源投入後に一度は原点復帰動作を実行する必要があります。
原点復帰未完了の状態で多機能電動ハンドを動作させるとエラーが発生しますが、ロボット本体はプログラムの実行などによって動作させることができます。

⚠ 注意

ロボットの動作時、多機能電動ハンドの開閉状態に起因してワークや周辺機器などとの干渉が起こる恐れがあります。
このような干渉を防止するため、多機能電動ハンドの状態を確認しながらロボットを動作させるよう、十分に注意して操作やプログラミングをおこなってください。
(多機能電動ハンドの状態をプログラムで確認するための状態変数を用意しています)
以下に、原点復帰完了状態をチェックするプログラムの例を示します。(状態変数と命令語の詳細は 561 ページの「7.2.8 ロボット言語の仕様」に記載しています)
また、ロボットがワークを把持する動作開始前にはハンドが開いていることを確認する必要があります。

【例 1】: 原点復帰を T/B などの操作でおこなう場合

[機能] プログラム起動時に多機能電動ハンド 1 の原点復帰が未完了なら、エラー 9100 を発生させる。

```
1 If M_EHOrg = 0 THEN      ' 原点復帰が未完了だったら ( 状態変数 M_EHOrg の値が "0")
2    Error 9100            '  ｴﾗｰ 9100 を発生させる。
3 EndIf
4  以下、作業プログラム ' ｽﾃｯﾌﾟ 1 で原点復帰が完了していれば本ｽﾃｯﾌﾟ以降を実行する。
5    :
6    :
```

補足 : 発生させるエラーは 428 ページの「UER1 ～ UER20」パラメータで設定しておきます。

【例 2】：原点復帰をプログラムの実行でおこなう場合

［機能］プログラム実行中に多機能電動ハンド 1 の原点復帰が未完了状態になったら、エラー 9100 を発生させる。

```
 1 EHOrg 1                                  ’原点復帰を実行する。
 2 Def Act 1, M_EHOrg = 0 GoSub *ORGLOST    ’割込み処理を定義する（原点復帰が未完了状態
                                              となったﾗﾍﾞﾙ：*ORGLOST へ処理を移す）
 3 Wait M_EHOrg = 1                         ’原点復帰完了を待つ。
 4 Act 1 = 1                                ’割込み処理を有効にする。
 5 以下、作業プログラム
 6    :                                     ’（実行中に原点復帰が未完了状態となったら
 7    :                                     ’ﾗﾍﾞﾙ：*ORGLOST へ処理が移る）
      :
 999 *ORGLOST                               ’
1000 Error 9100                             ’ｴﾗｰ 9100 を発生させる。
1001 Return 0                               ’割込んだ時のｽﾃｯﾌﾟへ戻る。
```

補足：①発生させるエラーは 428 ページの「UER1 ～ UER20」パラメータで設定しておきます。
②プログラム実行中に原点復帰が未完了となった場合、その原因となるアラーム（モータ過負荷など）が発生しますが、上記プログラムを実行することで 2 重の安全策をとることができます。

(2) ALWAYS（常時実行）プログラムでの実行について

各電動ハンド命令を ALWAYS プログラムで実行することができます。ALWAYS プログラムで実行可能にするため、特に設定するものはありません。

表 7-5：電動ハンド制御命令（概要）

命令	機能
EHOpen	ハンドの開
EHClose	ハンドの閉
EHMov	指定位置へ移動
EHHold	位置決め把持動作
EHOrg	原点復帰
EHStop	ハンド停止
EHTbl	テーブル動作命令

ALWAYS プログラムで実行した場合の注意点

1) ALWAYS(常時実行) プログラムで電動ハンド命令を使用していても、ALWAYS プログラムのみが実行している場合は運転中の出力信号は OFF のままです。
2) 電動ハンドのエラーに (二重実行、過負荷、フィードバックエラーなど) が発生してもプログラムは動き続けます。M_Err 変数の値をチェックして、エラー発生中は命令を実行させないプログラミングが可能です。
3) 非常停止を ON しても、電動ハンド命令を使用した ALWAYS(常時実行) プログラムは停止することなく動き続けます。M_Err 変数の値をチェックして、エラー発生中は命令を実行させないプログラミングが可能です。
4) 電源 ON 直後 (操作パネルがオーバーライド表示に変わったタイミング) から約 2 秒間は電動ハンドに初期データを送信しておりますので、その間に電動ハンドの各命令を実行すると二重実行のエラーが発生します。M_EHBusy が 0 になるのを待ってから実行してください。

ALWAYS プログラムの例

```
1  While M_EHBusy(1)=0              ' 電源 ON 時の初期化完了を待つ
2  If M_Err=0 Then                  ' エラー未発生時のみ電動ハンドの処理をする
3    If M_EHOrg(1)=0 Then           ' 原点復帰未の場合は、原点復帰実行
4      EHOrg 1
5      Wait M_EHOrg(1)=1
6    EndIf
7    If M_In(20)=1 Then             ' 入力信号 20 番がオンでハンド開
8      EHOpen 1,50,30               ' ハンドを開く
9    EndIf
10   If M_In(21)=1 Then             ' 入力信号 21 番がオンでハンド閉
11     EHClose 1,50,30              ' ハンドを閉じる
12   EndIf
13   Wait _EHBusy(1)=0              ' ハンドの処理完了を待つ
14 EndIf
15 End
```

7.2.5 操作手順

以下に多機能電動ハンドの操作手順概要を示します。

＜操作の流れ＞

(1) 多機能電動ハンドの取付けと接続 … 多機能電動ハンドに付属の取扱説明書を参照し、多機能電動ハンドをロボット本体のメカニカルインタフェースに取り付けます。
信号ケーブル（多機能電動ハンドに付属）を接続します。
注）取付方向は位置決めピンを合わせた方向に取り付けてください。

↓

(2) パラメータの設定 … ご使用になる多機能電動ハンドを正しく制御するために各パラメータを設定します。

↓

(3) 原点復帰 … 多機能電動ハンドを動作させるために基準位置を設定します。電源 ON 後には必ず必要です。

↓

(4) ハンド開閉、教示操作など … 上記設定の完了で多機能電動ハンドの操作が可能になります。

7.2.6 パラメータの設定

多機能電動ハンドの制御に必要なパラメータの設定をおこないます。
設定は、T/B(R32TB/R33TB、R56TB/R57TB) または RT ToolBox2 で設定します。パラメータ変更方法の詳細はそれぞれの製品に付属の取扱説明書を参照願います。R56TB/R57TB、および RT ToolBox2 には専用画面があります。

- R32TB/R33TB　：本書「3.14 パラメータ画面の操作」
- R56TB/R57TB　：R56TB/R57TB 取扱説明書 (BFP-A8591)
- RT ToolBox2　：RT ToolBox2 / RT ToolBox2 mini 取扱説明書 (BFP-A8617)

【注意】①ご使用になるハンド番号（ロータリースイッチによる局番設定）に対応するパラメータ名を正しく指定してください。
本書ではパラメータ名中のハンド番号に対応する文字を "n" で示しています。
＜例＞アクチュエータタイプ：EHnTYPE
ハンド番号 1 のアクチュエータタイプを設定する場合、パラメータ名は "EH1TYPE" と指定します。
②アクチュエータタイプを設定すると、制御パラメータは初期化されます。
手動で変更したパラメータの値は、ロボットの入替などの保守時に備えて、オプションの RT ToolBox2 を使用してバックアップを取っておくことをお勧めします。
③パラメータを書き込む時は、必ず電動ハンドの信号ケーブルを接続してください。

(1) アクチュエータタイプの設定

ご使用になる多機能電動ハンドのタイプを設定します。ご使用になる多機能電動ハンドに付属の取扱説明書を参照して設定してください。
表 7-6 に設定するパラメータの内容を示します。

表 7-6：アクチュエータタイプ設定パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
アクチュエータタイプ	EHnTYPE	整数 1	多機能電動ハンド本体のタイプ(機種)を設定します。 設定範囲は0～65535です。多機能電動ハンド本体のタイプ(電動ハンド形名)ごとの設定値を表7-7に示します。 n:ハンド番号(1～3)を示します。	0

注）本パラメータを設定すると、多機能電動ハンド本体から制御パラメータを読み出してコントローラのパラメータに自動で設定されます。（初期値）　それまでに変更されていたパラメータは上書きされますので、事前にオプションの RT ToolBox2 を使用してバックアップを取っておくことをお勧めします。

表 7-7：電動ハンド形名ごとの設定値

電動ハンド形名	設定値	電動ハンド形名	設定値	電動ハンド形名	設定値	電動ハンド形名	設定値
SS-2010	110	SS-2005-3N	111	SS-2005-5N	112	ST-2013	118
ST-2004	119	SS-2815	120	ST-2820	129	SS-4225	130
ST-4230	139	SD-2005	210	SD-2810	220	SD-4220	230
FS(T)2020	310	FS(T)2840	320				

(2) 制御パラメータの設定

表 7-8 に多機能電動ハンドを制御するために準備しているパラメータの一覧を、また表 7-12 にはそれぞれの詳細を示します。
必要に応じて変更してください。
なお、変更にあたっては多機能電動ハンドに付属の取扱説明書を参照願います。

表 7-8：制御パラメーター覧

No	パラメータ	パラメータ名 注 1)	配列数 文字数	内容説明	参照ページ
1	ソフトリミット＋側	EHnLMTP	整数 1	多機能電動ハンドのソフトリミット＋側 [0 ～ 9999] 0.1mm 単位	576
2	ソフトリミット－側	EHnLMTM	整数 1	多機能電動ハンドのソフトリミット－側 [-9999 ～ 0] 0.1mm 単位	576
3	ストローク	EHnSTRK	整数 1	多機能電動ハンドのストローク [0 ～ 9999] 0.1mm 単位	576
4	原点復帰方向	EHnORGD	整数 1	多機能電動ハンドの原点復帰方向 [0：OPEN/1：CLOSE]	577
5	原点復帰速度	EHnORGV	整数 1	多機能電動ハンドの原点復帰速度 [プログラム速度 20 ～ 50]	577
6	原点シフト	EHnORGSF	整数 1	多機能電動ハンドの原点シフト [-9999 ～ 9999] 0.01mm 単位	577
7	加速度	EHnACC	整数 1	多機能電動ハンドの加速度 [1 ～ 100] %	577
8	プログラム最高速度	EHnVMAX	整数 1	多機能電動ハンドのプログラム最高速度 [1 ～ 100] %	577
9	把持速度	EHnVHLD	整数 1	多機能電動ハンドの把持速度 [1 ～ 100] %	577
10	定速移動距離	EHnZNCV	整数 1	多機能電動ハンドの定速移動距離 [1 ～ 9999] 0.01mm 単位	577
11	リミット幅	EHnLMTW	整数 1	多機能電動ハンドのリミット幅 [1 ～ 9999] 0.01mm 単位	577
12	位置決め完了距離	EHnPSCD	整数 1	多機能電動ハンドの位置決め完了距離 [1 ～ 9999] 0.01mm 単位	576
13	原点復帰方式	EHnORGS	整数 1	多機能電動ハンドの原点復帰方式 [0：ストローク端 /1：ストローク端 +Z 相検出方向]	577
14	ポイントデータ	EHnPOS1 ～ EHnPOS32	整数 1	多機能電動ハンドのポイントデータ設定 [-999.99 ～ 999.99] 0.01mm 単位	577
15	テーブルデータ	EHnTBL1 ～ EHnTBL32	実数 7	多機能電動ハンドの動作条件テーブルデータ	577
16	T/B 動作モード	EHOPEMD	整数 1	TB からの動作モード初期値 [0：ポイント指定 /1：テーブル指定]	578
17	現在位置取得モード	EHCURMD	整数 1	TB への現在位置表示 [0：常時表示、1：MANUAL モードは常時表示、AUTOMATIC モードは動作終了で 1 回表示]	578

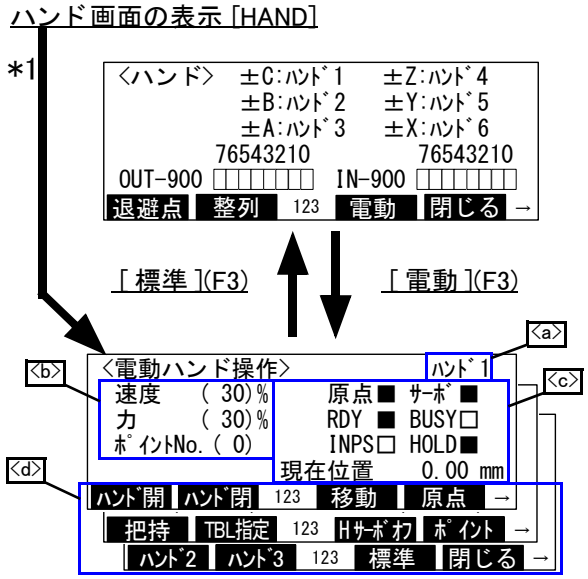
注 1）n：ハンド番号 (1 ～ 3) を示します。

7.2.7 多機能電動ハンドの操作

ここでは R32TB を使った多機能電動ハンドの操作方法を記載します。R56TB/R57TB での操作は別冊の「取扱説明書 /R56TB/R57TB 取扱説明書 (BFP-A8591)」を参照願います。

図 7-7 に R32TB での多機能電動ハンド操作画面を示します。

ハンド画面の表示 [HAND]



*1) 多機能電動ハンドが接続されている場合は最初に〈電動ハンド操作〉画面を表示します。

〈a〉
現在対象となっている多機能電動ハンド番号を表示します。
対象ハンドの切替えは〝ハンド n〟(n はハンド番号 1 ～ 3) に対応するファンクションキー [F1]、または [F2] キーを押します。

〈b〉
多機能電動ハンド動作のデータを設定します。
・速度 ..... 速度を % で設定します [1 ～ 100]
・力 ..... 力 (把持力) を % で設定します [1 ～ 100]
・ポイント No、またはテーブル No
※ ファンクションキー〝PNT 指定〟、〝TBL 指定〟で切り替わります。
・ポイント No ..... 教示位置番号を指定します [1 ～ 32]
・テーブル No ..... テーブル番号を指定します [1 ～ 32]

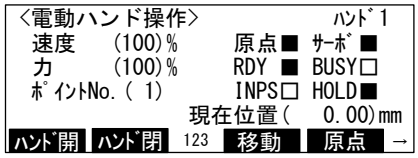
〈c〉
多機能電動ハンドの状態を表示します。
・原点 ..... ■ : 原点復帰完了 / □ : 未完了
・ｻｰﾎﾞ ..... ■ : ｻｰﾎﾞ ON / □ :OFF
・RDY ..... ■ : 準備完了 / □ : 準備中 / エラー発生中
・BUSY ..... ■ : 動作中 / □ : 動作中でない
・INPS ..... ■ : 目標位置到達 / □ : 未到達
・HOLD ..... ■ : 把持中 / □ : 把持していない
・現在位置 ..... 符号付絶対位置 (0.01mm 単位)

〈d〉
表示されている各ファンクションに対応するファンクションキー ([F1] ～ [F4]) を押して操作します。
(希望するファンクションが表示されていない場合は [FUNCTION] キーを押してください)
・〝ﾊﾝﾄﾞ開〟..... ハンドを開きます。(キーを離すと途中で停止します)
・〝ﾊﾝﾄﾞ閉〟..... ハンドを閉じます。(キーを離すと途中で停止します)
・〝移動〟 ..... 〝ﾎﾟｲﾝﾄ No〟に表示されている教示位置に向かって動作します。(キーを離すと途中で停止します)
・〝原点〟 ..... 原点復帰をおこないます。(キーを離すと途中で停止します)
・〝把持〟、または〝動作〟(※ 選択する毎に切り替わります。)
　・〝把持〟..... ハンド把持動作をおこないます。(キーを離すと途中で停止します)
　・〝動作〟..... 指定されたテーブル番号で設定された条件に基づいて動作します。
・〝TBL 指定〟、または〝PNT 指定〟(※ 選択する毎に切り替わります。)
　・〝TBL 指定〟..... テーブル番号 (動作させる条件) を指定します。(画面の 〈b〉 に〝テーブル No〟が表示されます)
　　注) テーブル番号を指定する場合、速度、力の条件はテーブルで指定された条件が有効になります。
　・〝PNT 指定〟..... 教示位置番号を指定します。(画面の 〈b〉 に〝ポイント No〟が表示されます)
・〝H ｻｰﾎﾞｵﾌ〟..... 多機能電動ハンドの駆動電源をオフします。
　※ 人手で移動させてダイレクトティーチが可能です。
・〝ﾎﾟｲﾝﾄ〟..... 多機能電動ハンドの位置を教示する画面に移ります。
・〝ﾊﾝﾄﾞn〟..... 操作の対象とする多機能電動ハンド番号を切替えます。(〝n〟はハンド番号を示します)
　注) 存在しない多機能電動ハンド番号は表示されません。
・〝標準〟 ..... 〈ハンド〉画面に戻ります。
・〝閉じる〟 ..... 〈電動ハンド操作〉画面、および〈ハンド〉画面を閉じ、元の画面に戻ります。

図 7-7 : 多機能電動ハンド操作画面

以下に各機能の操作方法を示します。ここでは〈電動ハンド操作〉画面を表示した状態での操作方法を説明します。多機能電動ハンドを事前に接続しておき、TB イネーブルスイッチを押して TB を有効状態にして、[HAND] キーを押して〈電動ハンド操作〉画面を表示させてください。

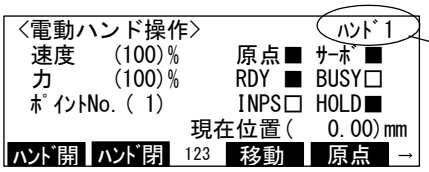
ハンド画面の表示 : [HAND] →



[注意] 外部入力信号の停止入力 (STOP、STOP2) が ON している時は、ハンドは動作しません。また、ハンド動作中に停止入力信号が ON すると直ちに停止します。

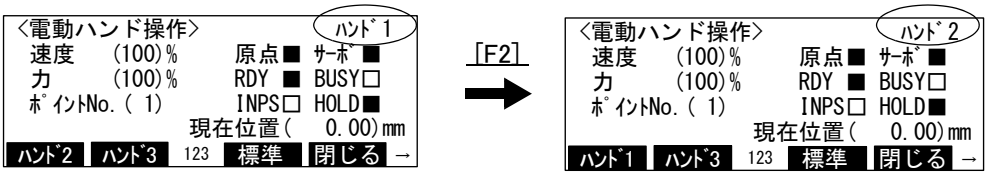
(1) ハンド番号の切替え

1) < 電動ハンド操作 > 画面の右上に、現在対象となっているハンド番号を表示しています。
ハンド番号が異なっている場合は、以下にしたがって切り替えてください。



対象となっているハンド番号

2) 対象とするハンド番号を切り替えるには、″ ハンド n″(n は 1 ～ 3) に対応するファンクションキーを押して切り替えます。



対象ハンド番号の切替え : [F1]、[F2]

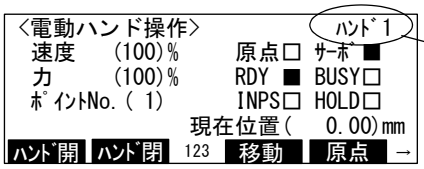
注) 図はハンド番号 1 から 2 に切り替えた例です。
″ ハンド n″ が表示されていない場合は、[FUNCTION] キーを押してください。

以上で対象とするハンド番号の切り替えは完了です。

(2) 原点復帰

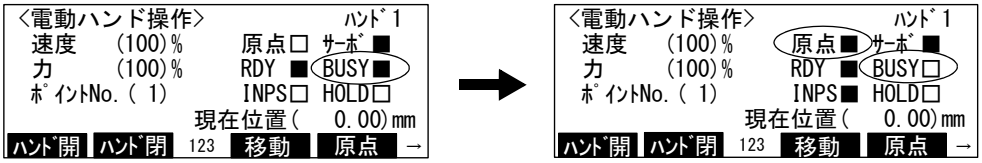
原点復帰とは、ハンドを動作させるための基準位置を設定する動作で、ハンドを動作させる前に必ず実施する必要があります。

1) 原点復帰の対象となるハンド番号を確認してください。(< 電動ハンド操作 > 画面右上の表示)
異なっている場合は切り替えてください。



対象となっているハンド番号を確認
注) 異なっている場合は 555 ページの「(1) ハンド番号の切替え」を参照 して切り替えてください。

2) < 電動ハンド操作 > 画面の ″ 原点 ″ に対応するファンクションキー ([F4]) を押し続けます。
キーを押している間、多機能電動ハンドは原点復帰動作を実行します。実行中は TB 画面の ″BUSY″ 表示が ″ ■ ″ になります。
原点復帰が完了すると、″BUSY″ 表示が ″ □ ″ となり、″ 原点 ″ 表示が ″ ■ ″ に変わります。



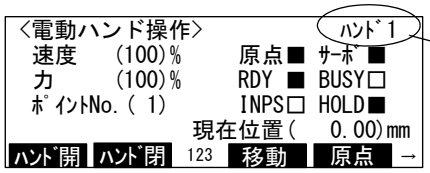
原点復帰 : [F4]
＊キーを離すと停止します。

原点復帰完了

以上で多機能電動ハンドの原点復帰操作は完了です。

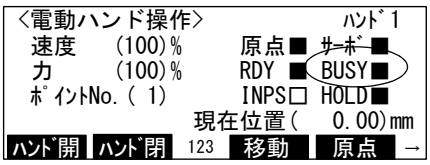
(3) ハンド開閉操作

1) ハンド開閉の対象となるハンド番号を確認してください。(< 電動ハンド操作 > 画面右上の表示)
異なっている場合は切り替えてください。



対象となっているハンド番号を確認
注) 異なっている場合は 555 ページの「(1) ハンド番号の切替え」を参照
して切り替えてください。

2) < 電動ハンド操作 > 画面の " ハンド開 "、" ハンド閉 " に対応するファンクションキーを押している間ハンドが動作します。動作中は "BUSY" 表示が " ■ " になります。



ハンドを開く　:" ハンド開 "([F1])
ハンドを閉じる :" ハンド閉 "([F2])

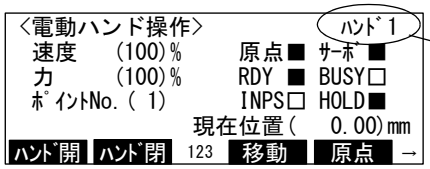
＊キーを離すと停止します。

3) ハンド開 / 閉端へ移動完了すると、"INPS" 表示が " ■ " になります。
注) 移動の途中で把持した場合は、"HOLD" 表示が " ■ " になります。

(4) 動作位置教示

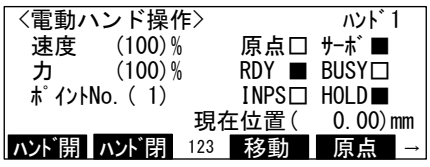
多機能電動ハンドの位置を教示しておき、プログラム (EHMov 命令) や TB で移動させることができます。
多機能電動ハンド 1 から 3 のそれぞれに、最大 32 点を教示することができ、番号で指定します。教示は多機能電動ハンドの現在位置を登録する方法の他に、直接数値を入力して登録することもできます。
以下に手順を示します。

1) 教示の対象となるハンド番号を確認してください。(< 電動ハンド操作 > 画面右上の表示)
異なっている場合は切り替えてください。



対象となっているハンド番号を確認
注) 異なっている場合は 555 ページの「(1) ハンド番号の切替え」を参照
して切り替えてください。

2) 多機能電動ハンドを、" ハンド開 "、" ハンド閉 " 等の操作により教示させる位置へ移動させてください。

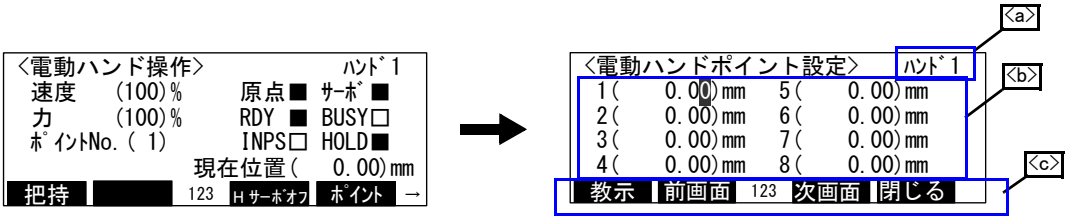


<u>教示位置への移動</u>
·ハンドを開く　:" ハンド開 "([F1])
·ハンドを閉じる :" ハンド閉 "([F2])

＊キーを離すと停止します。

·駆動電源をオフして、直接手で移動させても教示することができます。

3) ＜電動ハンド操作＞画面の〝ポイント〟に対応するファンクションキー([F4])を押して、＜電動ハンドポイント設定＞画面を表示させます。



＜電動ハンドポイント設定＞画面の表示：[F4]

<a>
教示の対象となっている多機能電動ハンド番号を表示します。

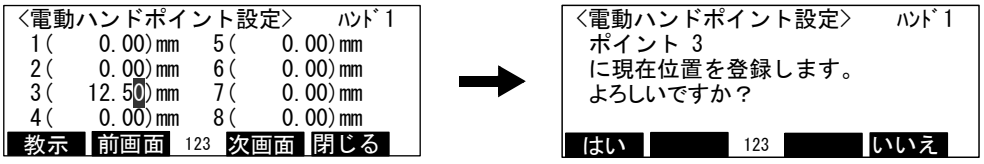
<b>
位置データを表示しています。
矢印キーでカーソルを移動させ数値入力による設定（微調整）が可能です。

<c>
・〝教示〟..... カーソル位置の位置データを教示します。
ハンドの現在位置が対象位置データに設定されます。
・〝前画面〟..... キーを押す毎に対象位置データを 1-8 ⇒ 25-32 ⇒ 17-24 ⇒ 9-16 ⇒ 1-8 に切替えます。
・〝次画面〟..... キーを押す毎に対象位置データを 1-8 ⇒ 9-16 ⇒ 17-24 ⇒ 25-32 ⇒ 1-8 に切替えます。
・〝閉じる〟.....＜電動ハンドポイント設定＞画面を閉じて＜電動ハンド操作＞画面へ戻ります。

4) 位置を教示します。
教示には以下に示すように a) 多機能電動ハンドの現在位置を登録する方法と b) 数値を入力して登録する方法があります。

a) 多機能電動ハンドの現在位置を登録する

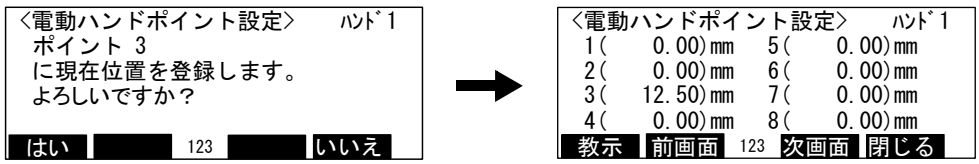
①矢印キーを押して、カーソルを教示するポイント番号へ移動させます。希望するポイント番号が表示されていない場合は、〝前画面〟、または〝次画面〟に対応するファンクションキーを押して表示させます。
〝教示〟に対応するファンクションキー([F1])を押します。確認画面が表示されます。



現在位置の教示：[F1]

注）図はハンド番号 1 のポイント 3 に現在の位置を教示する例です。

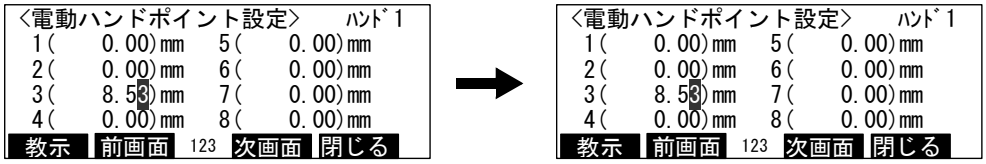
②〝はい〟に対応するファンクションキー([F1])を押すと現在位置が登録されます。
〝いいえ〟に対応するファンクションキー([F4])を押すとキャンセルします。



教示の実行：[F1]、
キャンセルは[F4]

以上で多機能電動ハンドの現在位置が登録されました。

b) 数値を入力して登録する
①矢印キーを押して、カーソルを教示するポイント番号へ移動させます。希望するポイント番号が表示されていない場合は、″前画面″、または″次画面″に対応するファンクションキーを押して表示させます。
直接数値を入力して [EXE] キーを押します。

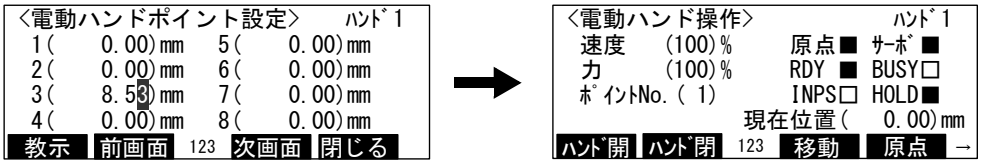


数値入力による教示：[数字] キー、[EXE]

注) 図はハンド番号 1 のポイント 3 を ″8.53″mm に設定する例です。

以上で数値入力により位置データが登録されました。

5) 多機能電動ハンドポイント設定画面を終了します。
″閉じる″に対応するファンクションキー ([F4]) を押すと〈電動ハンド操作〉画面に戻ります。

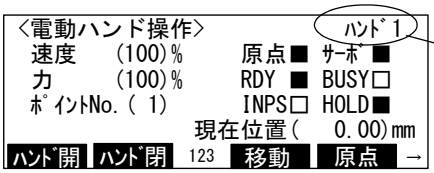


〈電動ハンドポイント設定〉画面の終了：[F4]

以上で多機能電動ハンドの動作位置教示は終了です。

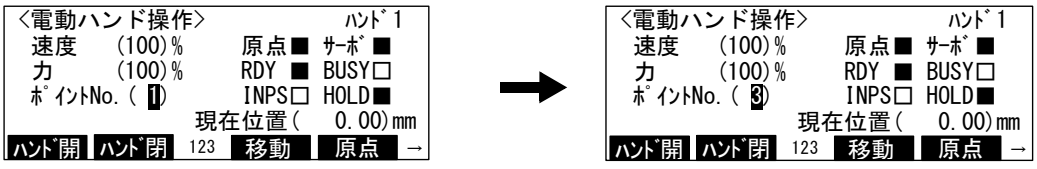
(5) 教示位置への移動
教示済みの位置へ多機能電動ハンドを移動させることができます。

1) 移動の対象となるハンド番号を確認してください。(〈電動ハンド操作〉画面右上の表示)
異なっている場合は切り替えてください。



対象となっているハンド番号を確認
注) 異なっている場合は 555 ページの「(1) ハンド番号の切替え」を参照 して切り替えてください。

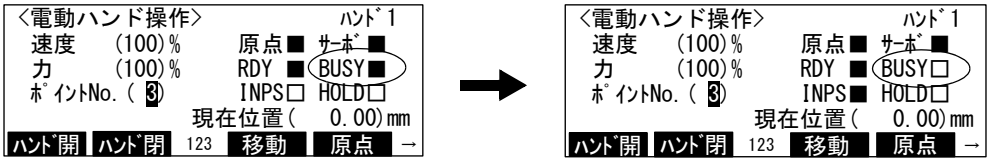
2) 画面上の″ポイント No.″に、移動させるポイント番号を指定します。
矢印キーでカーソルを″ポイント No.″のカッコ内に移動させ、数字 (1 ～ 32) を入力して [EXE] キーを押します。



カーソル移動：[矢印]、数値入力 [数字]

注) 図はハンド番号 1 のポイント 3 を指定した例です。

3) 画面上の〝移動〟に対応するファンクションキー([F3])を押し続けます。押している間、ハンドは指定されたポジション番号へ移動します。移動中は〝BUSY〟表示が〝■〟になります。
目標位置への移動が完了すると、〝BUSY〟表示が〝□〟となり、〝INPS〟表示が〝■〟になります。



指定ポジションへの移動：[F3]　　移動完了

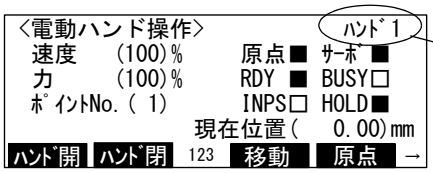
以上で教示位置への移動は終了です。

(6) 把持動作

教示済みの位置での把持動作を実行します。台形速度、定速移動距離などの把持動作パターンはパラメータの設定値に基づきます。

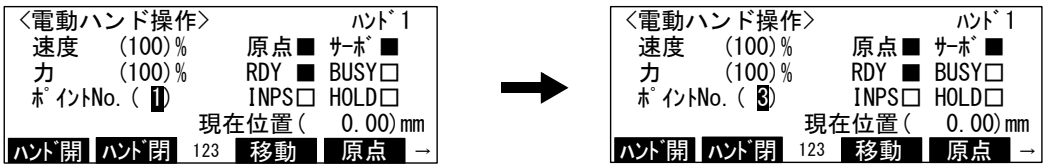
以下に把持動作の操作方法を示します。

1) 動作の対象となるハンド番号を確認してください。(〈電動ハンド操作〉画面右上の表示)
異なっている場合は切り替えてください。



対象となっているハンド番号を確認
注)異なっている場合は 555 ページの「(1) ハンド番号の切替え」を参照して切り替えてください。

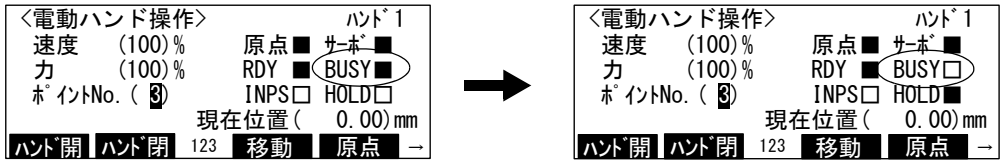
2) 画面上の〝ポイント No.〟に、把持させるポイント番号を指定します。
矢印キーでカーソルを〝ポイント No.〟のカッコ内に移動させ、数字(1 ～ 32)を入力して[EXE]キーを押します。



カーソル移動：[矢印]、数値入力[数字]

注)図はハンド番号 1 のポイント 3 を指定した例です。

3) 画面上の〝把持〟に対応するファンクションキー([F1])を押し続けます。押している間、ハンドは指定されたポジション番号での把持動作します。移動中は〝BUSY〟表示が〝■〟になります。



指定ポジションでの把持動作：[F1]　　把持動作完了

4) ワークを把持すると、〝BUSY〟表示が〝□〟、〝HOLD〟表示が〝■〟になります。
注)ワークを把持せず目標位置まで移動すると、〝INPS〟表示が〝■〟になります。

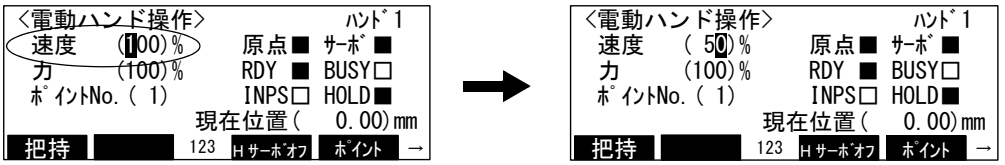
以上で教示位置での把持動作は終了です。

(7) 動作速度の設定

T/B から多機能電動ハンドを動作させる時の速度を設定することができます。設定値は最高速度に対する割合 (%) です。この設定値はハンド番号間で共通となります。

以下に設定操作方法を示します。

1) 矢印キーでカーソルを〝速度〟のカッコ内に移動させて、数値を入力して [EXE] キーを押します。



速度の設定 : [矢印]、[数字]、[EXE]　　設定完了

注) 図は速度を 50% に設定した例です。

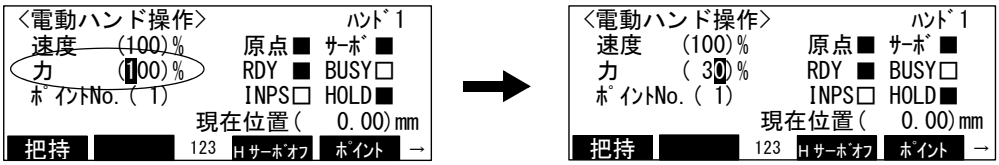
以上で動作速度の設定は終了です。

(8) 力 (把持力) の設定

T/B から多機能電動ハンドを動作させる時の力 (把持力) を設定することができます。設定値は最高把持力に対する割合 (%) です。この設定値はハンド番号間で共通となります。

以下に設定操作方法を示します。

1) 矢印キーでカーソルを〝力〟のカッコ内に移動させて、数値を入力して [EXE] キーを押します。



力 (把持力) の設定 : [矢印]、[数字]、[EXE]　　設定完了

注) 図は力を 30% に設定した例です。

注) 設定値が小さい場合、ハンド自身の摩擦によって動作しないことがあります。
例) 多機能電動ハンド : ESG1-SS-2815-11XW107 では、10% 以上で動作します。

以上で力の設定は終了です。

7.2.8 ロボット言語の仕様

(1) 命令語一覧

表 7-9 に多機能電動ハンドを動作させる命令語の一覧を示します。

表 7-9：多機能電動ハンド制御命令

No	命令語	解説	参照ページ
1	EHOpen/EHClose（ハンドオープン / ハンドクローズ）	ストローク開端までの動作 / ストローク閉端までの動作	562
2	EHMov（ハンドムーブ）	指定ポジションへの位置決め動作	564
3	EHHold（ハンドホールド）	指定ポジションでの位置決め把持動作	566
4	EHOrg（ハンドオリジン）	原点復帰動作	568
5	EHServo On/Off（ハンドサーボ）	サーボモータ電源の ON/OFF を制御	569
6	EHStop（ハンドストップ）	動作の停止	570
7	EHTbl（ハンドテーブル）	指定されたテーブル番号の条件で動作	571

(2) 状態変数一覧

表 7-10 に多機能電動ハンドに関係する状態変数の一覧を示します。

表 7-10：多機能電動ハンド状態変数

No	変数名称	配列要素数注 1)	内容	属性注 2)	データの型、単位	参照ページ
1	M_EHPos1/ M_EHPos2/M_EHPos3	ポイント番号(1 ～ 32)	多機能電動ハンド別(1～3)のポイントデータ	RW	単精度実数型、mm	572
2	M_EHCur	ハンド番号(1 ～ 3)	多機能電動ハンドの現在位置	R	単精度実数型、mm	573
3	M_EHBusy	ハンド番号(1 ～ 3)	多機能電動ハンドの動作状態 [1:実行中/0:非実行中]	R	整数型	574
4	M_EHInPs	ハンド番号(1 ～ 3)	多機能電動ハンドの目標位置到達状態 [1:目標位置到達/0:未到達]	R	整数型	574
5	M_EHHold	ハンド番号(1 ～ 3)	多機能電動ハンドの把持検出状態 [1:把持検出/0:未検出]	R	整数型	574
6	M_EHZone	ハンド番号(1 ～ 3)	多機能電動ハンドの指定領域内 [1:指定領域内/0:指定領域外]	R	整数型	574
7	M_EHOrg	ハンド番号(1 ～ 3)	多機能電動ハンドの原点復帰完了状態 [1:原点復帰完了/0:未完了]	R	整数型	574
8	M_EHTMd1/ M_EHTMd2/ M_EHTMd3	テーブル番号(1 ～ 32)	多機能電動ハンドの指定されたテーブルデータの動作モード	R	整数型	575
9	M_EHOPos	ハンド番号(1 ～ 3)	多機能電動ハンドの動作完了ポイント番号 [0:原点未完了、又は移動中]	R	整数型	576

注 1) ポイント番号.....1 ～ 32、ポイントデータ（教示番号）を指定します。
ハンド番号.....1 ～ 3、ハンド番号を指定します。
テーブル番号.....1 ～ 32、ハンドのテーブル番号を指定します。

注 2) R.....読み出しのみ可能です。
RW.....読み出し、書き込みとも可能です。

(3) 命令語の詳細説明
　　各命令後の詳細を機能別に示します。

EHOpen/EHClose（ハンドオープン／ハンドクローズ）

【機能】
　　多機能電動ハンドの開 / 閉を動作をおこないます。

【書式】

EHOpen 口＜ハンド番号＞,＜速度＞,＜力＞[,＜ゾーン ON 位置＞,＜ゾーン OFF 位置＞]
EHClose 口＜ハンド番号＞,＜速度＞,＜力＞[,＜ゾーン ON 位置＞,＜ゾーン OFF 位置＞]

【用語】
　　＜ハンド番号＞ .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。
　　（注:小数点以下は四捨五入されます）
　　＜速度＞.....ハンドの移動速度を%単位の整数で指定します。定数または数値変数で指定します。
　　設定範囲:1～100(%) （注:小数点以下は四捨五入されます）
　　注）①設定速度が20%より小さいと、ハンドが振動する場合があります。
　　②設定速度が50%より大きいと、過負荷エラーが発生することがあります。
　　＜力＞.....ハンドの把持力を%単位を整数で指定します。定数または数値変数で設定します。
　　設定範囲:1～100(%) （注:小数点以下は四捨五入されます）
　　注)設定値が30%より小さい場合、ハンド自身の摩擦により動作しないことがあります。
　　例)多機能電動ハンド:ESG1-SS-2815-11XW107では、10%以上で動作します。
　　＜ゾーンON位置＞/＜ゾーンOFF位置＞
　　ハンドが特定領域内かどうかを確認する場合、その特定領域を先頭位置/最終位置で指定します。(図7-8参照) 定数または数値変数で指定します。
　　設定範囲:-9999.99～9999.99(mm) (0.01mm単位)

【文例 1】（ゾーン指定なし）

```
1 EHOpen 1, 100, 30     ' ハンド 1 を開く（速度 =100%, 力 =30%）
2 Mov PUP               ' 把持位置上空へ移動
3 Wait M_EHInPs=1       ' 開端への移動完了確認
4 Mvs PGET              ' 把持位置へ移動
5 Fine 0.5, P           ' ロボットの移動完了を待つ 注 1)
6 EHClose 1, 50, 30     ' ハンド 1 を閉じる（速度 =50%, 力 =30%）
7 Wait M_EHBusy=0       ' ハンドの動作完了を待つ
8 If M_EHHold=1 Then    ' もし把持していたら、把持位置上空へ移動
9   Mvs PUP
10 EndIf
```

注 1)Fine 命令を使用することで、ロボット本体の動作が完了してからハンドの動作を始めることができます。但し、Fine 命令で指定するパルス数 (0.5) は調整を要する値です。(217 ページの「Fine（ファイン）」参照）
　　また、Dly 命令でも同様の効果を得ることができます。

【文例 2】(ゾーン指定あり)

```
1 Mvs PCHK                              ' 寸法チェック位置へ移動
2 Fine 0.5, P                           ' ロボットの移動完了を待つ 注 1)
3 EHClose 1, 50, 30, 20.00 22.00        ' ハンド 1 を閉じる
                                          ( 速度 =50%, 力 =30%, ゾーン位置 =20 ～ 22mm)
4 Wait M_EHBusy=0                       ' ハンドの動作完了を待つ
5 If M_EHHold=1 AND M_EHZone=1 Then     ' もしゾーン位置で把持していたら MCHKOK( チェックフラグ )
                                          に "1" を代入。
6   MCHKOK=1
7 EndIf
8 EHMov 1, 0, 100, -10.00               ' ハンド 1 を現在位置から 10mm 開く ( 速度 =100%)
9 Wait M_EHInPs=1                       ' 指定位置への移動完了確認
10 Mvs PUP                              ' 寸法チェック位置上空へ移動
```

注 1)Fine 命令を使用することで、ロボット本体の動作が完了してからハンドの動作を始めることができます。但し、Fine 命令で指定するパルス数 (0.5) は調整を要する値です。(217 ページの「Fine (ファイン)」参照)
また、Dly 命令でも同様の効果を得ることができます。

【解説】

1) 速度、力を指定してハンドを開 / 閉します。
一定速度で開、または閉方向にストローク端まで移動します。ストローク途中で把持を検出した場合はその位置で動作を完了します。

2) 自動運転中、本命令はハンド動作を開始させたら次の命令実行に進みます。(ハンド動作の終了を待ちません)

a) ハンドの動作完了を待ってからロボットを移動させる必要がある場合は、以下の状態変数を参照してください。参照方法を上記文例に示しています。
- M_EHBusy(ハンド動作中)..... ハンド動作中 "1"
- M_EHInPs(目標位置到達状態)..... 動作開始時に "0"、開、または閉端に到達したら "1"
- M_EHHold(ハンド把持状態)..... 動作開始時に "0"、ストローク途中で把持したら "1"

b) ハンド動作中、ハンド動作命令を実行したスロットが停止したらハンド動作も一時停止します。スロットが再起動したらハンド動作も再開されます。
ただしスロット停止中に T/B の手動操作等で対象ハンドを動作させた場合は、ハンド動作は再開しません。

c) プログラムからハンド動作を途中で停止 (終了) させる場合は、EHStop 命令を実行ください。

3) ステップ運転もしくはダイレクト実行では本命令の実行でハンド動作の終了まで待ちます。

4) ゾーン位置を指定して停止位置確認をすることもできます。
ハンドがゾーン内にいるときは状態変数 M_EHZone が "1" になります。

例)EHClose 1, 50, 30, 5, 8 を実行した場合。(ﾊﾝﾄﾞの位置が 5mm ～ 8mm では M_EHZone の値が "1" になります)

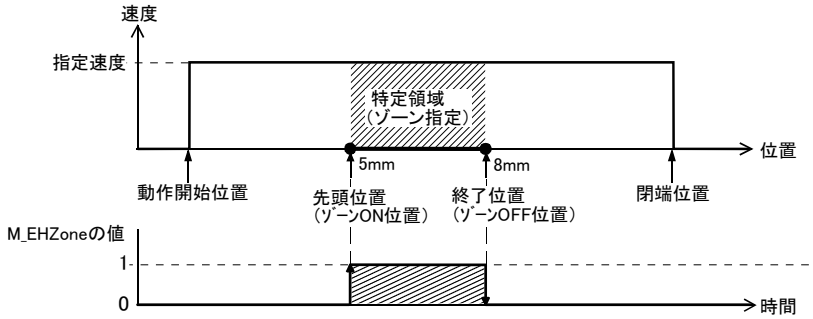


図 7-8 : ハンド動作と状態変数 (M_EHZone) の値 (例)

5) 多機能電動ハンド動作中に本命令を実行した際はエラー「ハンド動作命令二重実行」が発生します。
二重実行の可能性がある場合は状態変数 M_EHBusy(ハンド動作中) を参照してハンド動作中でないことを確認してから本命令を実行ください。

6) コントローラ電源立上げ後は、多機能電動ハンドを原点復帰させてから本命令を実行してください。
原点復帰せずに本命令を実行した場合はエラーが発生します。

7) 多機能電動ハンドはご購入後、パラメータ初期化 (機種選択) していただく必要があります。

8) コンティニュー機能には対応していません (動作中に電源を落とすと処理が初期化されます)。

EHMov（ハンドムーブ）

【機能】

多機能電動ハンドを指定位置へ移動させます。

【書式】

EHMov □＜ハンド番号＞,＜ポイント番号＞,＜速度＞[,＜相対量＞]

【用語】

＜ハンド番号＞.....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。
（注:小数点以下は四捨五入されます）

＜ポイント番号＞.....多機能電動ハンドのポイント番号を指定します。定数または数値変数で指定します。
設定範囲は0～32です。（注:小数点以下は四捨五入されます）
0は現在位置、1～32はパラメータ:EHnPOS1～32(n:ハンド番号)に設定された値を目的位置に設定します。

＜速度＞.....多機能電動ハンドの移動速度を%単位の整数で指定します。定数または数値変数で指定します。
設定範囲:1～100(%)（注:小数点以下は四捨五入されます）
注）設定速度が20%より小さいと、ハンドが振動する場合があります。

＜相対量＞.....この値を指定した場合、実際の移動目的位置は本設定値分オフセットした位置となります。
設定値は0.01mm単位で指定します。定数または数値変数で指定します。
設定範囲は-9999.99～9999.99(mm)です。(注:0.01mm以下は四捨五入されます)

【文例 1】（ゾーン指定なし）

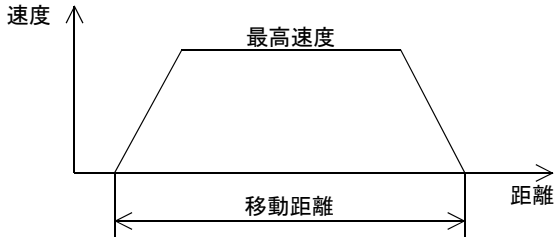
```
1 EHMov 1, 5, 100            ' ハンド 1 移動（ポイント番号 =5, 速度 =100%）
2 Mov PUP                    ' 把持位置上空へ移動
3 Wait M_EHInPs=1            ' 指定位置への移動完了確認
4 Mvs PGET                   ' 把持位置へ移動
5 Fine 0.5, P                ' ロボットの移動完了を待つ（EHOpen/EHClose（ハンドオープン / ハンドクローズ）の文例参照）
6 EHClose 1, 50, 30          ' ハンド 1 閉（速度 =50%, 力 =30%）
7 Wait M_EHBusy=0            ' 動作終了確認
8 If M_EHHold=1 Then         ' 把持確認
9   Mvs PUP                  ' 把持位置上空へ移動
10 EndIf
```

【文例 2】（ゾーン指定あり）

```
1 Mvs PCHK                              ' 寸法チェック位置へ移動
2 EHClose 1, 50, 30, 20.00 22.00        ' ハンド 1 閉（速度 =50%, 力 =30%, ゾーン位置 =20 ～ 22mm）
3 Wait M_EHBusy=0                       ' 動作終了確認
4 If M_EHHold=1 AND M_EHZone=1 Then     ' 把持 & ゾーン確認
5   MCHKOK=1                            ' チェックフラグ =1
6 EndIf
7 EHMov 1, 0, 100, -10.00               ' ハンド 1 移動：現在位置から 10mm 開（速度 =100%）
8 Wait M_EHInPs=1                       ' 指定位置への移動完了を待つ
9 Mvs PUP                               ' 寸法チェック位置上空へ移動
```

【解説】

1) ポイント番号、速度を指定して多機能電動ハンドを指定位置へ移動させます。
指定位置までは台形速度制御で移動します。



2) 自動運転中、本命令はハンド動作を開始させたら次の命令実行に進みます。(ハンド動作の終了を待ちません)
a) ハンドの動作完了を待ってからロボットを移動させる必要がある場合は、以下の状態変数を参照してください。参照方法を上記文例に示しています。
・M_EHBusy(ハンド動作中)……ハンド動作中 "1"
・M_EHInPs(目標位置到達状態)……動作開始時に "0"、指定位置に到達したら "1"
b) ハンド動作中、ハンド動作命令を実行したスロットが停止したらハンド動作も一時停止します。
スロットが再起動したらハンド動作も再開されます。
ただしスロット停止中に T/B の手動操作等で対象ハンドを動作させた場合は、ハンド動作は再開しません。
c) プログラムからハンド動作を途中で停止 (終了) させる場合は、EHStop 命令を実行ください。
3) ステップ運転もしくはダイレクト実行では本命令の実行でハンド動作の終了まで待ちます。
4) 相対量を指定すると、移動位置を相対量分オフセットさせることができます。
5) 移動位置へ行く途中で外力により停止した場合はエラー「多機能電動ハンドユニットアラーム (モータ過負荷)」が発生します。
6) 多機能電動ハンド動作中に本命令を実行した際はエラー「ハンド動作命令二重実行」が発生します。
二重実行の可能性がある場合は状態変数 M_EHBusy(ハンド動作中) を参照してハンド動作中でないことを確認してから本命令を実行ください。
7) コントローラ電源立上げ後は、多機能電動ハンドを原点復帰させてから本命令を実行してください。
原点復帰せずに本命令を実行した場合はエラーが発生します。
8) 多機能電動ハンドはご購入後、パラメータ初期化 (機種選択) していただく必要があります。
9) コンティニュー機能には対応していません (動作中に電源を落とすと処理が初期化されます)。

EHHold（ハンドホールド）

【機能】
多機能電動ハンドの位置決め把持動作を実施します。

【書式】

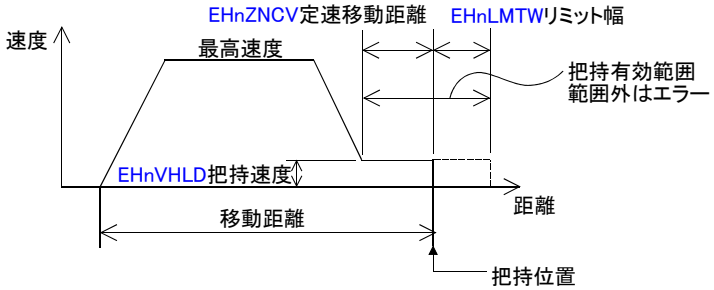
EHHold □＜ハンド番号＞,＜ポイント番号＞,＜速度＞,＜力＞

【用語】
＜ハンド番号＞ .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。
注)実数の数値変数で指定した場合は四捨五入して設定します。
＜ポイント番号＞.....多機能電動ハンドのポイント番号を指定します。定数または数値変数で指定します。
設定範囲は0～32です。（注:小数点以下は四捨五入されます）
0は現在位置、1～32はパラメータ:EHnPOS1～32(n:ハンド番号)に設定された値を目的位置に設定します。
＜速度＞.....ハンドの移動速度を%単位の整数で指定します。
設定範囲:1～100(%)
注）設定速度が20%より小さいと、ハンドが振動する場合があります。
＜力＞.....ハンドの把持力を%単位を整数で指定します。定数または数値変数で設定します。
設定範囲:1～100(%)
注)①実数の数値変数で指定した場合は四捨五入して設定します。
②設定値が30%より小さい場合、ハンド自身の摩擦により動作しないことがあります。
例)多機能電動ハンド:ESG1-SS-2815-11XW107では、10%以上で動作します。

【文例1】

```
1 EHOpen 1, 100, 30                  ' ハンド1開（速度=100%, 力=30%）
2 Mov PUP                            ' 把持位置上空へ移動
3 Wait M_EHInPs=1                    ' 開端への移動完了確認
4 Mvs PGET                           ' 把持位置へ移動
5 Fine 0.5, P                        ' ロボットの移動完了を待つ（EHOpen/EHClose（ハンドオープン/ハンドクローズ）の文例参照）
6 EHHold 1, 5, 100, 30               ' ハンド1位置決め把持（ﾎﾟｲﾝﾄ番号=5, 速度=100%, 力=30%）
7 Wait M_EHBusy=0                    ' 動作終了確認
8 If M_EHHold=1 AND M_EHZone=1 Then  ' 把持＆ゾーン確認
9   Mvs PUP                          ' 把持位置上空へ移動
10 EndIf
```

【解説】
1) ポイント番号、速度、力を指定して多機能電動ハンドの位置決め把持動作を実施します。
指定位置手前まで台形速度制御で移動し把持直前で定速動作し把持します。
把持直前の定速移動距離(EHnZNCV)・把持速度(EHnVHLD)・指定位置を越えた有効距離（リミット幅)(EHnLMTW)はパラメータで設定します。



2) 自動運転中、本命令はハンド動作を開始させたら次の命令実行に進みます。(ハンド動作の終了を待ちません)
   a) ハンドの動作完了を待ってからロボットを移動させる必要がある場合は、以下の状態変数を参照してください。参照方法を上記文例に示しています。
      ・M_EHBusy(ハンド動作中)……ハンド動作中 "1"
      ・M_EHHold(ハンド把持状態)……動作開始時に "0"、ストローク途中で把持したら "1"
      ・M_EHZone(指定領域停止状態)……指定領域 (定速移動距離 + リミット幅内) にいるとき "1"
      ・M_EHInPs(目標位置到達状態)……動作開始時に "0"、目標位置に到達したら "1"
   b) ハンド動作中、ハンド動作命令を実行したスロットが停止したらハンド動作も一時停止します。
      スロットが再起動したらハンド動作も再開されます。
      ただしスロット停止中に T/B の手動操作等で対象ハンドを動作させた場合は、ハンド動作は再開しません。
   c) プログラムからハンド動作を途中で停止 (終了) させる場合は、EHStop 命令を実行ください。
3) ステップ運転もしくはダイレクト実行では本命令の実行でハンド動作の終了まで待ちます。
4) 定速移動距離とリミット幅の範囲内で把持した場合は状態変数 M_EHHold と M_EHZone が 1 にセットされます。
   定速移動距離手前で把持した場合は状態変数 M_EHHold のみ 1 にセットされます。
   リミット幅の範囲内で把持できなかった場合は M_EHHold は 0 で停止し、M_EHInPs を 1 にします。
5) 動作途中で停止した場合は、再始動の動作で移動開始位置に戻ってから再度位置決め把持動作を実施します。
   移動開始位置に戻る動作は EHMov 命令 (速度は EHHold 命令で指定された速度) 動作とします。
6) 多機能電動ハンド動作中に本命令を実行した際はエラー「ハンド動作命令二重実行」が発生します。
   二重実行の可能性がある場合は、状態変数 M_EHBusy(ハンド動作中) を参照して、ハンド動作中でないことを確認してから本命令を実行ください。
7) コントローラ電源立上げ後は、多機能電動ハンドを原点復帰させてから本命令を実行してください。
   原点復帰せずに本命令を実行した場合はエラーが発生します。
8) 多機能電動ハンドはご購入後、パラメータ初期化 (機種選択) していただく必要があります。
9) コンティニュー機能には対応していません (動作中に電源を落とすと処理が初期化されます)。

EHOrg（ハンドオリジン）

【機能】
多機能電動ハンドの原点復帰を実施します。

【書式】

EHOrg □＜ハンド番号＞

【用語】
＜ハンド番号＞ .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。
注)実数の数値変数で指定した場合は四捨五入して設定します。

【文例 1】
```
1 If M_EHOrg=0 THEN      ' 原点復帰未完了状態
2   EHOrg 1              ' ハンド 1 原点復帰
3   Wait M_EHOrg=1       ' ハンド 1 原点復帰完了待ち
4 EndIf
5 EHOpen 1, 100, 30      ' ハンド 1 開（速度 =100%, 力 =30%）
```

【解説】
1) 多機能電動ハンドを原点復帰させます。
コントローラ電源立上げ後、多機能電動ハンドの移動動作をさせる前には原点復帰させる必要があります。
2) 自動運転中、本命令はハンド動作を開始させたら次の命令実行に進みます。（ハンド動作の終了を待ちません）
a) ハンドの動作終了確認は以下の状態変数を参照しておこなってください。
・M_EHBusy（ハンド動作中）.....ハンド動作中 "1"
・M_EHInPs（目標位置到達状態）.....動作開始時に "0"、原点復帰が完了したら "1"
・M_EHOrg（ハンド原点復帰完了状態）.....原点復帰が完了している時 "1"
b) ハンド動作中、ハンド動作命令を実行したスロットが停止したらハンド動作も一時停止します。
スロットが再起動したらハンド動作も再開されます。
ただしスロット停止中に T/B の手動操作等で対象ハンドを動作させた場合は、ハンド動作は再開しません。
c) プログラムからハンド動作を途中で停止（終了）させる場合は、EHStop 命令を実行ください。
3) 原点復帰させた際に、前回の原点復帰時の位置が再現されない場合はエラー「多機能電動ハンドユニットアラーム（エンコーダ Z 相位置ずれ）」が発生します。
その際には位置ずれの要因を取り除いた後、再度原点復帰を実施ください。
爪交換などにより原点位置が変わった場合は、多機能電動ハンドのパラメータ初期化（機種選択）操作を実施ください。
4) 多機能電動ハンド動作中に本命令を実行した際はエラー「ハンド動作命令二重実行」が発生します。
二重実行の可能性がある場合は状態変数 M_EHBusy（ハンド動作中）を参照してハンド動作中でないことを確認してから本命令を実行ください。
5) 多機能電動ハンドはご購入後、パラメータ初期化（機種選択）していただく必要があります。
6) コンティニュー機能には対応していません（動作中に電源を落とすと処理が初期化されます）。
7) 本命令を使用せず T/B などで原点復帰をおこなう場合、以下に示す原点復帰完了チェック機能をプログラムの先頭にを記述しておくと、原点復帰が完了していない時、事前にエラーを発生させることができます。
原点復帰完了チェック機能（例：原点復帰が未完了だったら、エラー 9100 を発生させる）
```
1 If M_EHOrg = 0 THEN   '
2   Error 9100
3 EndIf
```
補足：発生させるエラーは 428 ページの「UER1 ～ UER20」パラメータで設定しておきます。

⚠ 注意　多機能電動ハンドの原点設定動作中は、ロボットを動作させないでください。ロボットを動作させると、振動などの影響により原点設定が正しく完了しない場合があります。

EHServo On/Off（ハンドサーボ）

【機能】

多機能電動ハンドのサーボモータ電源の ON/OFF を制御します。

【書式】

EHServo □＜ハンド番号＞,＜On/Off＞

【用語】

＜ハンド番号＞ .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。
注)実数の数値変数で指定した場合は四捨五入して設定します。

＜On/Off＞ .....On:多機能電動ハンドのサーボモータの電源を入れる場合
Off:多機能電動ハンドのサーボモータの電源を切る場合

【文例】

```
1  EHMov 1, 1, 100          'ハンド1移動（ポイント番号=1、速度=100%）
2  Mov PUP                  '把持位置上空へ移動
3  Wait M_EHBusy=0          '指定位置への移動完了確認
4  EHServo 1, Off           'ハンド1サーボOFF
5  Mvs P1                   'ハンド1の爪やワークがテーパ形状で、爪がワークに接触した場合、ワークに倣いハンド1の爪が動きます。（シングルカム・三爪タイプのみ）
```

【解説】

1) 多機能電動ハンドのサーボモータ電源の制御をおこないます。
※EHOpen/EHClose/EHMov/EHHold/EHOrg 命令の実行時には自動的にサーボモータ電源が入りますので、EHServo On を実行する必要はありません。
※ サーボモータ電源を切ると、外部からの力で可動部を動かすことができます。（シングルカム・三爪タイプのみ）
2) 本命令はサーボモータ電源の ON/OFF が完了してから次の命令実行に移行します。
3) 多機能電動ハンド動作中に本命令を実行させるとエラー「ハンド動作命令二重実行」が発生する場合があります。このような場合は、状態変数 M_EHBusy（ハンド動作中）を参照して、ハンド動作中でないことを確認してから、本命令を実行してください。

EHStop（ハンドストップ）

【機能】

多機能電動ハンドを停止させます。

【書式】

EHStop □＜ハンド番号＞

【用語】

＜ハンド番号＞ .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。
注)実数の数値変数で指定した場合は四捨五入して設定します。

【文例 1】

```
1  EHClose 1, 100, 30                ' ハンド 1 閉（速度 =100%, 力 =30%）
2 *LOOP: If M_In(20)=1 Then EHStop ' 入力 20 番（ｾﾝｻ情報など）が ON したらﾊﾝﾄﾞ動作を途中停止
3  If M_EHBusy=1 Then Goto *LOOP    ' 動作中はループ
4  Mov PUP                           ' 上空位置へ移動
```

【解説】

1) 多機能電動ハンドを停止させます。
EHOpen/EHClose/EHMov/EHHold/EHOrg 命令により実行された動作を停止させます。
2) 多機能電動ハンドはご購入後、パラメータ初期化（機種選択）していただく必要があります。
3) 本命令実行後、直ちにハンド動作命令(EHOpen など）を実行させるとエラー「ハンド動作命令二重実行」が発生する場合があります。このような場合は、状態変数 M_EHBusy(ハンド動作中）を参照して、ハンド動作中でないことを確認してから、ハンド動作命令を実行してください。

EHTbl（ハンドテーブル）

【機能】

多機能電動ハンドを指定されたテーブル番号の条件で動作させます。

【書式】

EHTbl □＜ハンド番号＞, ＜テーブル番号＞

【用語】

＜ハンド番号＞ .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。

＜テーブル番号＞ .....多機能電動ハンドのテーブル番号を指定します。
指定範囲は1～32です。
パラメータはEHnTBL1～32に設定されたテーブル情報が設定されます。

【文例 1】

```
1  EHTbl 1, 1                  ' ハンド 1 移動（ポイント番号 =5、速度 =100%、相対量 =-3.0mm）
2  Mov PUP                     ' 把持位置上空へ移動
3  Wait M_EHInPs=1             ' 開端への移動完了確認
4  Mvs PGET                    ' 把持位置へ移動
5  Fine 0.5, P                 ' 移動完了を待つ（Dly 命令でも OK です）
6  EHTbl 1, 2                  ' ハンド 1 位置決め把持（ポイント番号 =5、速度 =100%、力 =30%）
7  Wait M_EHBusy=0                          ' 動作終了確認
8  If M_EHHold=1 AND M_EHZone=1 Then        ' 把持 & ゾーン確認
9    Mvs PUP                                ' 把持位置上空へ移動
10 EndIf
```

＜パラメータ設定＞

```
EH1TBL1=3.00, 5.00, -3.00, 100.00, 0.00, 0.00, 0.00
EH1TBL2=4.00, 5.00, 0.00, 100.00, 30.00, 0.00, 0.00
```

【解説】

1) 多機能電動ハンドをテーブル指定実行させます。
パラメータ EHnTBL1 ～ 32 に設定されたテーブル情報（動作モード、ポイント番号、相対量、力、ゾーン ON 位置、ゾーン OFF 位置）に従い、多機能電動ハンドを動作させます。

2) 自動運転中、本命例はハンド動作を開始させたら次の命令実行に移行します（ハンド動作の終了を待ちません）。
ハンドの動作終了確認は以下の状態変数を参照しておこなってください。
＜動作モード：開、閉の場合＞
- M_EHBusy（ハンド動作中）.....ハンド動作中 "1"
- M_EHInPs（目標位置到達状態）.....動作開始時に "0"、移動が完了したら "1"
- M_EHHold（ハンド把持状態）.....動作開始時に "0"、ストローク途中で把持したら "1"

(4) 状態変数の詳細説明
各状態変数の詳細を機能別に示します。

M_EHPos1/M_EHPos2/M_EHPos3

【機能】
多機能電動ハンド 1 ～ 3 のポイントデータを参照、設定します。

【書式】

```
<数値変数>=M_EHPos1(<ポイント番号>)      'ポイントデータの参照
M_EHPos1(<ポイント番号>)=<数値データ>     'ポイントデータの設定
```

【用語】
<ハンド番号> .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。
注)実数の数値変数で指定した場合は四捨五入して設定します。
<数値変数> .....ポイントデータを代入する変数を指定します。
ポイントデータ値は-9999.99～9999.99(mm)です。
<数値データ>.....ポイントデータを指定します。定数または数値変数で指定します。
設定値は0.01mm単位で指定します。
設定範囲は-9999.99～9999.99(mm)です。(注:0.01mm以下は四捨五入されます)

【文例】

```
1 MPos=M_EHCur          'ハンド現在位置を代入
2 MPos=MPos+0.5         '現在位置 +0.5mm の位置を算出
3 M_EHPos1(5)=MPos      'ポイント 5 へ設定
4 EHMov 1, 5, 100       'ポイント 5 の位置へハンド 1 移動 ( ポイント番号 =5, 速度 =100%)
5 Wait M_EHInPs=1       '指定位置への到達確認
```

【解説】
1) ポイント番号を指定して多機能電動ハンドのポイントデータ (パラメータ EHnPOS1 ～ EHnPOS32、n : ハンド番号) を設定・参照します。

M_EHCur

【機能】
多機能電動ハンドの現在位置を返します。

【書式】

<数値変数>=M_EHCur [(<ハンド番号>)]

【用語】
<数値変数> .....ハンド現在位置を代入する変数を指定します。
<ハンド番号> .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。(省略時はハンド1です)
注)実数の数値変数で指定した場合は四捨五入して設定します。

【文例】

```
1 MPos=M_EHCur          'ハンド現在位置を代入
2 MPos=MPos+0.5         '現在位置 +0.5mm の位置を算出
3 M_EHPos1(5)=MPos      'ポイント 5 へ設定
4 EHMov 1, 5, 100       'ポイント 5 の位置へハンド 1 移動 ( ポイント番号 =5, 速度 =100%)
5 Wait M_EHInPs=1       '指定位置への到達確認
```

【解説】
1) ハンド番号を指定して多機能電動ハンドの現在位置を返します。
2) 現在位置は -9999.99 ～ 9999.99(mm) の範囲になります。
3) 対象ハンドが接続されていないときの値は 0 になります。
4) 対象ハンドが原点復帰されていないときの値は不定です。

M_EHBusy
M_EHInPs
M_EHHold
M_EHZone
M_EHOrg

【機能】

多機能電動ハンド状態を返します。

M_EHBusy.....動作状態 [1：動作実行中 /0：動作実行なし]
M_EHInPs.....目標位置到達状態 [1：目標位置到達 /0：未到達]
M_EHHold.....把持状態 [1：把持検出 /0：未検出]
M_EHZone.....指定領域内 [1：指定領域内にいる /0：いない]
M_EHOrg .....原点復帰状態 [1：原点復帰完了 /0：未完了]

【書式】

< 数値変数 >=M_EHOrg [(< ハンド番号 >)] ※ 他の状態変数も同等です。

【用語】

<数値変数> .....多機能電動ハンド状態を代入する変数を指定します。
<ハンド番号> .....多機能電動ハンド番号1～3を整数で指定します。定数または数値変数で指定します。(省略時はハンド1です)
注)実数の数値変数で指定した場合は四捨五入して設定します。

【文例】

```
1 If M_EHOrg=0 THEN                    ' 原点復帰未完了状態
2   EHOrg 1                            ' ハンド 1 原点復帰
3   Wait M_EHOrg=1                     ' ハンド 1 原点復帰完了待ち
4 EndIf
5 EHOpen 1, 100, 20                    ' ハンド 1 開 ( 速度 =100%, 力 =20%)
6 Mov PUP                              ' 把持位置上空へ移動
7 Wait M_EHInPs=1                      ' 開端への移動完了確認
8 Mvs PGET                             ' 把持位置へ移動
9 Fine 0.5, P                          ' ロボットの移動完了を待つ (EHOpen/EHClose（ハンドオープン / ハンドクローズ）の文例参照 )
10 EHClose 1, 50, 20, 20.00 22.00      ' ﾊﾝﾄﾞ 1 を閉じる ( 速度 =50%, 力 =20%, ｿﾞｰﾝ位置 =20 ～ 22mm)
11 Wait M_EHBusy=0                     ' 動作終了確認
12 If M_EHHold=1 AND M_EHZone=1 Then   ' 把持 & ゾーン確認
13   Mvs PUP                           ' 把持位置上空へ移動
14 EndIf
```

【解説】

1) ハンド番号を指定して多機能電動ハンドの各種状態を返します。
2) 対象ハンドが接続されていないときは 0 になります。
3) 各状態変数の解説は以下の通りです。

表 7-11：多機能電動ハンドの状態変数

No	状態変数	解説
1	M_EHBusy	動作状態[1:動作実行中/0:動作実行なし]。 ハンド動作処理中には"1"が設定されます。
2	M_EHInPs	目標位置到達状態[1:目標位置到達/0:未到達]。 ハンド動作開始時に"0"となり、目標位置(位置決め完了距離範囲)に到達すると"1"になります。目標位置にいる間、"1"が設定されます。
3	M_EHHold	把持状態[1:把持検出/0:未検出] ハンド動作開始時に"0"となり、動作終了時に指定把持力で把持していたら"1"になります。指定把持力で把持している間、"1"が設定されます。

No	状態変数	解説
4	M_EHZone	指定領域内[1:指定領域内にいる/0:いない] ハンド動作開始時に"0"となり、指定領域に入ると"1"になります。 指定領域内にいる間、"1"が設定されます。
5	M_EHOrg	原点復帰状態[1:原点復帰完了/0:未完了] ハンド原点復帰動作が正常に完了したら"1"が設定されます。 コントローラ電源立上げ後は、原点復帰させないと各種動作命令は実行できません。 本状態変数より原点復帰完了状態を確認して、原点復帰未完了の場合は原点復帰命令(EHOrg)を実行させてください。

M_EHTMd1/M_EHTMd2/M_EHTMd3

【機能】

多機能電動ハンドの指定されたテーブルデータの動作モードを返します。

【書式】

```
<数値変数>=M_EHTMd1 [(<テーブル番号>)]      'テーブルデータの動作モードを参照
```

【用語】

<数値変数> .....テーブルデータの動作モード(0～5)を代入する変数を指定します。
0:未設定、1:移動、2:把持、3:相対移動、4:開(オープン)、5:閉(クローズ)
<テーブル番号> .....多機能電動ハンドのテーブル番号を指定します。
設定範囲は1～32です。定数または変数で設定します。

【文例】

```
M1=M_EHTMd1(1)              'ハンド1のテーブルデータ1の動作モードをM1へ代入
M11=M_EHTMd2(11)            'ハンド2のテーブルデータ11の動作モードをM11へ代入
If M11<>0 Then EHTbl 2, 11  'ハンド2をテーブルデータ11の内容で動作
```

【解説】

1) ハンド番号を指定して多機能電動ハンドのテーブルデータ(パラメータ EHnTBL1 ～ EHnTBL32、n:ハンド番号)の動作モードを設定・参照します。

M_EHOPos

【機能】
多機能電動ハンドの移動完了ポイント番号を返します。

【書式】

<数値変数>=M_EHOPos [(<ハンド番号>)]

【用語】
<数値変数> .....移動完了ポイント番号(1～32)を代入する変数を指定します。

【文例】
```
M1=MEHOPos(1)                    ' ハンド 1 の完了ポイント番号を M1 へ代入
If M_EHOPos(2)=2 Then *L01       ' ハンド 2 の完了ポイント番号が 2 であれば *L01 へジャンプ
```

【解説】
1) ハンド番号を指定して多機能電動ハンドの移動完了ポイント番号を返します。
2) EHOrg/EHOpen/EHClose/EHMov/EHHold の命令実行開始時に 0 となります。
3) EHMov の動作完了後に M_EHInPs=1 でポイント番号を出力します。
4) EHHold の動作完了後に M_EHHold=M_EHZone=1、または M_EHInPs=1 でポイント番号を出力します。
5) EHHold の把持有効範囲外での動作完了後の出力は 0 のままです。また、把持したワークが取り除かれた状態で把持動作をおこない M_EHInPs=1 となった場合も、出力は 0 のままです。
6) EHMov/EHHold の命令実行後、移動中にエラー発生もしくは中断した場合、出力は 0 のままです。
7) 対象ハンドが接続されていないときは 0 になります。

7.2.9 パラメーター覧

表 7-12 にパラメータの一覧を示します。

表 7-12：多機能電動ハンド制御パラメーター覧

パラメータ	パラメータ名 注 1)	配列数 文字数	内容説明	出荷時設定値
多機能電動ハンド アクチュエータタイプ	EHnTYPE	整数 1	多機能電動ハンド本体のタイプ(機種)選択の実施および記憶します。 設定範囲は0～65535です。 本パラメータ書込み時には接続された多機能電動ハンドのパラメータ初期化(機種選択)操作を実施した後、対象ハンドのパラメータ初期値を記憶します。	0
多機能電動ハンド ソフトリミット (+)	EHnLMTP	整数 1	多機能電動ハンドのプラス側の可動範囲を設定します。 設定範囲は0～9999(0.1mm単位)です。 原点復帰方向を開側へ設定した場合、閉側が+方向となります。 原点復帰方向を閉側へ設定した場合、開側が+方向となります。	0
多機能電動ハンド ソフトリミット (-)	EHnLMTM	整数 1	多機能電動ハンドのマイナス側の可動範囲を設定します。 設定範囲は-9999～0(0.1mm単位)です。 原点復帰方向を開側へ設定した場合、閉側が+方向となります。 原点復帰方向を閉側へ設定した場合、開側が+方向となります。	0
多機能電動ハンド ストローク	EHnSTRK	整数 1	多機能電動ハンドのアクチュエータのストロークを設定します。 設定範囲は0～9999(0.1mm単位)です。	0
多機能電動ハンド 位置決め完了距離	EHnPSCD	整数 1	指定したポイントへ移動する際に、指定した距離の手前で移動完了となります。 設定範囲は1～9999(0.01mm単位)です。 システムのタクトタイムを上げたい時に値を大きくします。	0

パラメータ	パラメータ名 注 1)	配列数 文字数	内容説明	出荷時設定値
多機能電動ハンド 加速度	EHnACC	整数 1	アクチュエータの加速度を設定します。 アクチュエータタイプと可動部質量の設定により最適な加速度が自動設定されます。 アクチュエータ設定部および爪部の剛性等を考慮して加速度を下げたい場合に変更を行ってください。 設定範囲は1～100(%)です。	0
多機能電動ハンド 最高速度	EHnVMAX	整数 1	最高速度を設定します 設定範囲は1～100(%)です。	0
多機能電動ハンド 把持速度	EHnVHLD	整数 1	位置決め把持動作(EHHold命令での動作)の把持有効範囲内での定速速度を設定します。 設定範囲は1～100(%)です。	0
多機能電動ハンド 定速移動距離	EHnZNCV	整数 1	位置決め把持動作(EHHold命令での動作)の定速移動時の距離を設定します。 目標位置より定速移動距離分だけ手前から把持速度になります。 設定範囲は1～9999(0.01mm単位)です。	0
多機能電動ハンド リミット幅	EHnLMTW	整数 1	位置決め把持動作(EHHold命令での動作)の定速移動時のリミット幅を設定します。 目標位置よりリミット幅分だけ、把持速度にて移動します。 把持有効範囲を超えた場合、移動量+リミット幅で停止します。 設定範囲は1～9999(0.01mm単位)です。	0
多機能電動ハンド 原点復帰方向	EHnORGD	整数 1	原点復帰方向を設定します。設定値は0:開側/1:閉側です。 原点復帰方向を「0:開側」へ設定した場合、位置設定が+で閉方向へ移動します。	0
多機能電動ハンド 原点復帰速度	EHnORGV	整数 1	原点復帰速度を設定します。設定範囲は20～50(%)です。	0
多機能電動ハンド 原点シフト	EHnORGSF	整数 1	原点復帰が完了した座標位置にこのパラメータで設定した値だけデータがシフトされます。 設定範囲は-9999～9999(0.01mm単位)です。 装置メンテナンス等で位置ずれが発生した場合、このパラメータ設定によりポイントデータの再入力の手間が省けます。ソフトリミットより大きな値は入力しないでください。	0
多機能電動ハンド 原点復帰方式	EHnORGS	整数 1	原点復帰の方法を設定します。 設定値は[0:ストローク端/1:ストローク端+Z相検出方式]です。 ストローク端検出後、Z相検出までフィンガは反転します。Z相検出までの反転距離が問題となる場合、ストローク端検出へ変更することにより反転距離が0.5mmとなります。	0
多機能電動ハンド ポイントデータ	EHnPOS1 ～ EHnPOS32	整数 1	多機能電動ハンドのポイントデータを設定(記憶)します。 設定範囲は-999.99～999.99(0.01mm単位)です。	0
多機能電動ハンド テーブルデータ	EHnTBL1 ～ EHnTBL32	実数 7	多機能電動ハンドのテーブルデータを設定します。 第1要素:動作モード (0:未設定、1:移動、2:把持、3:相対移動、 4:開(オープン)、5:閉(クローズ)) 第2要素:ポイント番号 〈設定範囲〉 動作モード/1:0～32/2:1～32 第3要素:相対量[mm] 〈設定範囲〉 -9999.99～9999.99 第4要素:速度[%] 〈設定範囲〉 動作モード/4,5:1～50/1,2,3:1～100 第5要素:力[%] 〈設定範囲〉 1～100 第6要素:ゾーンON位置[mm] ※動作モードが開(EHOpen)、閉(EHClose)のみ有効 〈設定範囲〉 -9999.99～9999.99 第7要素ゾーンOFF位置[mm] ※動作モードが開(EHOpen)、閉(EHClose)のみ有効 ※第6,7要素が両方とも"0"の時は、ゾーン指定は無効 〈設定範囲〉 -9999.99～9999.99	0,0,0,0,0,0,0

パラメータ	パラメータ名 注 1)	配列数 文字数	内容説明	出荷時設定値
多機能電動ハンド T/B 動作モード	EHOPEMD	整数 1	多機能電動ハンドのTB動作モードの初期値を設定します。 0:ポイント指定 1:テーブル指定	0
多機能電動ハンド 現在位置取得モード	EHCURMD	整数 1	多機能電動ハンドの現在位置取得モードを設定します。 0:常時取得します(従来互換) 1:Manualモード時は常時取得します。 Automaticモード時は動作終了時1回だけ取得します。	0

注 1) n：ハンド番号(1 ～ 3)を示します。

7.2.10 エラー一覧

表 7-13 に多機能電動ハンドに関するエラー一覧(原因と対策)を示します。
エラー番号の上 3 桁で対象となる多機能電動ハンド番号を区別しています。

- H814*：ハンド 1
- H815*：ハンド 2
- H816*：ハンド 3

表 7-13：エラー一覧表

エラー番号	エラー発生原因とその対策	
H8140 H8150 H8160	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンドリモート I/O 通信回線異常
	原因	多機能電動ハンドリモート I/O 通信で CRC エラー、または接続エラーが発生しました。
	対策	通信ケーブルのチェック、または接続機器の電源を確認してください。
L8141 L8151 L8161	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンドアラーム発生
	原因	多機能電動ハンドで、下記いずれかのアラームが発生しました。 モータ過負荷、モータ過電流、エンコーダ Z 相ずれ、電源電圧低下、位置偏差オーバー、フィードバックエラー 1，2，及び 3、電圧異常、システム異常 1，及び 2。
	対策	多機能電動ハンドの取扱説明書を参照してアラーム原因を取り除いた後、ロボットのエラーリセット操作をおこなってください。エラーリセット操作でエラーが解除されない場合はロボットの電源を入れ直してください。
L8142 L8152 L8162	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンドエラー発生
	原因	多機能電動ハンドで、下記いずれかのエラーが発生しました。 ソフトリミットオーバー、原点復帰未完了、駆動電源 OFF、インターロック時に移動命令を実行、運転中に実行命令を入力、データ書き込み中にコマンド入力、原点復帰時に Z 相が見つからない、FCS チェックエラー、データ入力範囲オーバー、アクチュエータタイプ間違い、内部通信異常。
	対策	多機能電動ハンドの取扱説明書を参照してエラー原因を取り除いた後、ロボットのエラーリセット操作をおこなってください。エラーリセット操作でエラーが解除されない場合はロボットの電源を入れ直してください。 プログラムで原点復帰の完了状態をチェックする例を 549 ページの「操作上の注意事項」に示しています。
L8143 L8153 L8163	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンド未接続
	原因	多機能電動ハンドが接続されていません。
	対策	リモート I/O の通信ケーブルの接続、または局番号の設定を確認してください。
L8144 L8154 L8164	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンド機種選択未完了。
	原因	多機能電動ハンドの機種選択が完了していません。
	対策	ロボットのパラメータ：アクチュエータタイプ (EHnTYPE) を設定してください。
L8145 L8155 L8165	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンド原点復帰未完了。
	原因	多機能電動ハンドの原点復帰が完了していません。
	対策	原点復帰を実行してください。 プログラムで原点復帰の完了状態をチェックする例を 549 ページの「操作上の注意事項」に示しています。
L8146 L8156 L8166	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンド応答タイムアウト。
	原因	多機能電動ハンドからの応答がありません。
	対策	ケーブルの接続状態、多機能電動ハンド状態を確認してコントローラの電源を入れ直してください。

エラー番号	エラー発生原因とその対策	
L8147 L8157 L8167	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンド動作命令二重実行。
	原因	多機能電動ハンドの動作命令を二重に実行しました。
	対策	動作命令を二重に実行しないようにプログラムを修正してください。
L8148 L8158 L8168	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンドｎポイント登録二重実行。
	原因	マルチタスクプログラムでの設定や、T/B と RT ToolBox2 などの操作でポイントデータ登録が重複しました。
	対策	プログラムの修正、または操作手順を見直してください。
L8149 L8159 L8169	ｴﾗｰﾒｯｾｰｼﾞ	多機能電動ハンドシステム異常 3。
	原因	多機能電動ハンドでシステム異常が発生しました。
	対策	エラーリセット後に原点復帰を実行してください。エラーがリセットできないときは、再度電源を投入してください。再発する場合はメーカまでお問い合わせください。
L0092	ｴﾗｰﾒｯｾｰｼﾞ	この信号は電動ハンドで使用中です。
	原因	電動ハンド用の出力信号は使用できません。
	対策	多機能電動ハンドが接続されている場合、多機能電動ハンドで使用する出力信号は、プログラムで出力することはできません。プログラムで使用する出力信号番号を変更してくださ い。

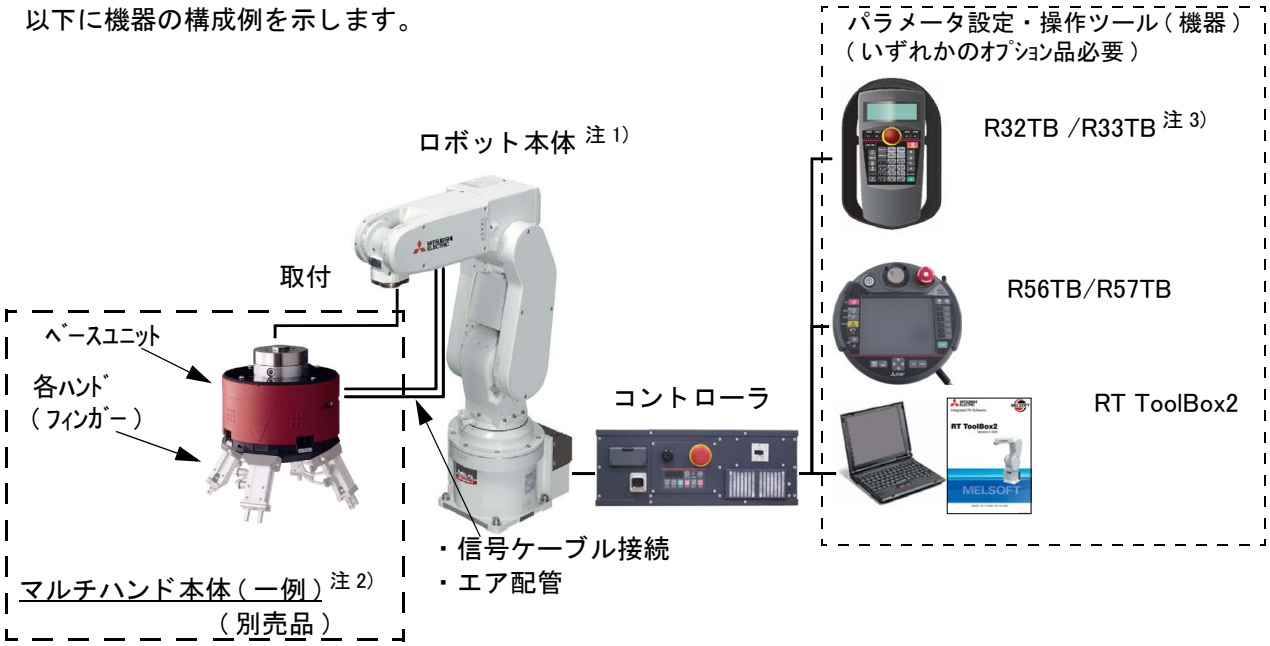
7.3 マルチハンドの使用方法

IDEC 社製マルチハンドをロボットの RIO に接続してご使用する際の設定、及び操作方法について説明します。マルチハンド製品に付属の取扱説明書も合わせてご参照ください。

7.3.1 概要

(1) 機器構成（例）

以下に機器の構成例を示します。



注 1) 図は RV-2F の構成例です。
RV-2F では標準仕様のロボットでマルチハンドのご使用が可能です。
他のロボットでは特殊仕様となります。（垂直多関節 6 軸タイプのみ）別途発行の納入仕様書を参照願います。
注 2) マルチハンド本体の図は一例を記載しています。マルチハンドを構成する付属品、および仕様等につきましてはマルチハンドに付属の取扱説明書を参照願います。
注 3) パラメータの設定操作を容易にするため、R56TB/R57TB、又は RT ToolBox2 をご使用願います。
ジョグ操作、ハンド開閉操作は R32TB/R33TB でも容易に操作できます。

図 7-9：マルチハンド使用時の構成例

【参考：マルチハンド本体とロボットで制御できるハンド（ﾌｨﾝｶﾞｰ）の関係】

マルチハンド本体は、ベースユニットと各ハンド（ﾌｨﾝｶﾞｰ）に分けて考えられます。ベースユニットは共通性がありますが、各ハンド（ﾌｨﾝｶﾞｰ）はワークに応じて形状が異なるのが一般的です。
本ロボットでは、各ハンド（ﾌｨﾝｶﾞｰ）基準での前進 / 後退など、その時に作業対象となっているハンド（ﾌｨﾝｶﾞｰ）基準で動作できるよう、ロボットの制御点をその都度ハンド（ﾌｨﾝｶﾞｰ）の先端（作業位置）に変更する機能を備えています。
お客様でのジョグ操作や自動運転でツール番号を指定すると、制御点を指定されたツール番号に対応するハンド（ﾌｨﾝｶﾞｰ）の先端に変更します。（制御点の初期値はメカニカルインタフェースの中心）
この機能を有効に活用するために、各ハンド（ﾌｨﾝｶﾞｰ）とメカニカルインタフェース中心との位置関係を予めパラメータに設定しておきます。（ツール番号に対応するハンド（ﾌｨﾝｶﾞｰ）の関係付け） 設定するパラメータの詳細を 581 ページの「(1) パラメータの設定」に記載します。

(2) 使用可能な機器とソフトウェアバージョン

マルチハンドは表 7-14 に示すロボットと機器、ツール、および対象のソフトウェアバージョンでご使用できます。

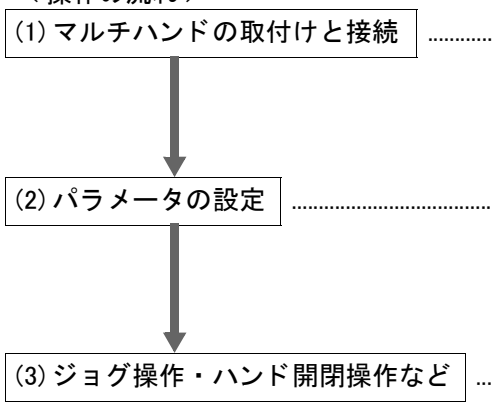
表 7-14：使用可能な機器とソフトウェアバージョン

機器	コントローラ	R32TB/R33TB	R56TB/R57TB	RT ToolBox2
使用可能なソフトウェアバージョン	R3 以降	Ver. 1.6 以降	Ver. 2.5 以降	Ver. 2.1 以降

7.3.2 操作手順

以下に操作手順の概要を示します。

＜操作の流れ＞



(1) マルチハンドに付属の取扱説明書を参照し、マルチハンドをロボット本体のメカニカルインタフェースに取り付けます。
信号ケーブル（マルチハンド本体と別売）、およびエア配管（お客様ご準備）を接続します。
注）取付方向は位置決めピンを合わせた方向に取り付けてください。

(2) ご使用になるマルチハンドを正しく制御するために各パラメータ（ハンドタイプ、初期状態、ツール長など）を設定します。
注）パラメータの設定には、R56TB/R57TB、または RT ToolBox2 をご使用願います。
R56TB/R57TB、または RT ToolBox2 をご使用いただくとツール座標パラメータ：MEXTLn(n：1 ～ 16) の設定値は自動設定されます。

(3) 上記設定の完了でマルチハンドを使用した操作が可能になります。

(1) パラメータの設定

パラメータは、T/B または RT ToolBox2 で設定します。操作方法の詳細はそれぞれの製品に付属の取扱説明書を参照願います。

- R32TB/R33TB：本書「7.3.2 操作手順」
- R56TB/R57TB：R56TB/R57TB 取扱説明書 (BFP-A8591)
- RT ToolBox2：RT ToolBox2/RT ToolBox2 mini 取扱説明書 (BFP-A8617)

1) ハンドタイプの設定

パラメータ：ハンドタイプ (HANDTYPE) を設定することで、T/B からのハンド操作や HOpen/HClose 命令で通常のハンドと同様な取扱いが可能となります。
マルチハンドを RIO で接続した場合、チャネル 1 の局番 5 を使用するため入出力信号番号は以下となります。

- 入力信号番号.....716 ～ 731(16 点)
- 出力信号番号.....716 ～ 723(8 点)

注）入出力とも 700 ～ 715 番はシステム予約領域のためご使用できません。
また、入力信号番号はマルチハンド製品の仕様に基づきます。

設定例）ハンド開閉をダブルソレノイドで以下の出力信号でおこなう場合の設定例を示します

ハンド 1 開 / 閉信号番号：716/717
ハンド 2 開 / 閉信号番号：718/719
ハンド 3 開 / 閉信号番号：720/721
ハンド 4 開 / 閉信号番号：722/723

パラメータ：HANDTYPE の設定
HANDTYPE = D716, D718, D720, D722, , , ,

表 7-15：ハンドタイプ設定パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
ハンドタイプ	HANDTYPE	文字列 8	シングル/ダブルソレノイドのハンドタイプと、出力信号番号を設定します(D:ダブルソレノイド、S:シングルソレノイド)。 ハンドタイプに続けて信号番号を設定します。 D900の場合、信号番号900と901に信号出力されます。 D(ダブルソレノイド)の場合は信号がかさならないように設定してください。	D900,D902,D904,D906 ,,,,

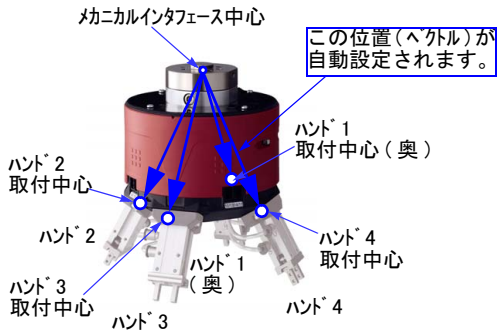
2) ハンドベースの形名設定

ご使用になるマルチハンドのベースの形名をパラメータ：MHBnTYPE(n：ﾍﾞｰｽ番号 1 ～ 8) に設定します。この設定値を基に、パラメータ：マルチハンドベース基準座標 (MHBnHCy。表 7-17) が自動で設定されます。

表 7-16：ハンドベース設定パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
マルチハンドベース形名	MHBnTYPE (n：ﾍﾞｰｽ番号 1 ～ 8)	文字列 16	ベース番号1～8に、ご使用になるマルチハンドベースの形名を割当(設定)します。 同時に8種類の形名が登録できます。 【以下出荷時設定値】 <ﾏﾙﾁﾊﾝﾄﾞ対応機種(垂直6軸特殊仕様を含む)> MHB1TYPE=MH1A-4NBY-* MHB2TYPE～MHB8TYPE=未設定 <上記以外の機種(含ﾕｰｻﾞﾒｶ)> MHB1TYPE～MHB8TYPE=未設定	左記参照

表 7-17：ハンド基準座標設定パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
マルチハンドベース基準座標	MHBnHCy (n：ﾍﾞｰｽ番号 1 ～ 8、 y：ﾊﾝﾄﾞ番号 1 ～ 4)	実数 6	ご使用になるマルチハンドベース毎の(ベース番号毎)、それぞれのハンド(ﾌｨﾝｶﾞｰ)取付中心位置を示します。 (ﾒｶﾆｶﾙｲﾝﾀﾌｪｰｽ中心からの相対位置) マルチハンドベース1個当たり4つのハンド(ﾌｨﾝｶﾞｰ)取付位置を持っています。 【以下出荷時設定値】 <ﾏﾙﾁﾊﾝﾄﾞ対応機種(垂直6軸特殊仕様を含む)> MHB1HC1=(0.00, +51.41, +107.41, 0.00, +30.00, +90.00) MHB1HC2=(+51.41, 0.00, +107.41, 0.00, +30.00, 0.00) MHB1HC3=(0.00, -51.41, +107.41, 0.00, +30.00, -90.00) MHB1HC4=(-51.41, 0.00, +107.41, 0.00, +30.00, +180.00) MHB2HC1～MHB8HC4=(0.00, 0.00, 0.00, 0.00, 0.00, 0.00) <上記以外の機種(含ﾕｰｻﾞﾒｶ)> MHB1HC1～MHB8HC4=(0.00, 0.00, 0.00, 0.00, 0.00, 0.00)  	左記参照

3) マルチハンドベースの割当 (設定)

同時に使用できる 4 種類のマルチハンドそれぞれに、使用するマルチハンドベース番号を割り当てます。

表 7-18：ハンドタイプ設定パラメータ

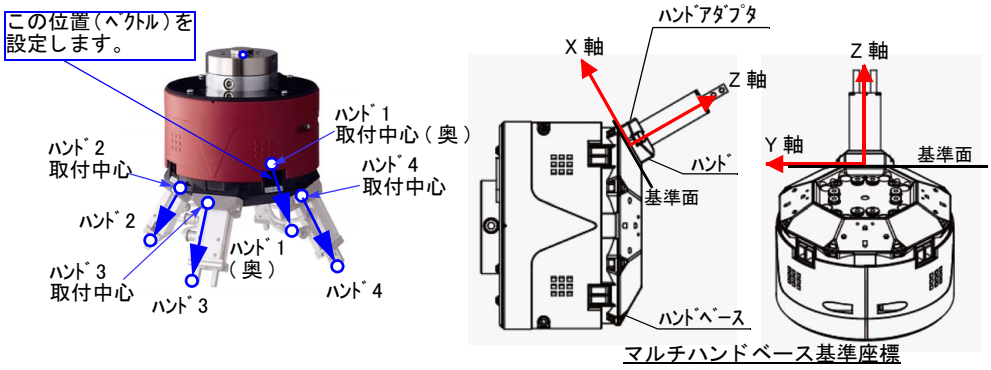
パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
マルチハンドベース番号	MHnBNO (n：ﾏﾙﾁﾊﾝﾄﾞ番号 1 ～ 4)	整数 1	マルチハンド番号nで使用する"マルチハンドベース番号"(パラメータ:MHBnTYPEの"n"に相当)を割り当てます。 設定値:0～8 0:未設定の状態にします。 1～8:割り当てるマルチハンドベース番号を指定します。 この設定値が、パラメータ:マルチハンドオフセット量(MHnOFSy)の各マルチハンド番号に対応します。	0

4) ハンドオフセット量の設定

4 種類のハンド（ﾌｨﾝｶﾞｰ）毎にその取付位置から作業位置（制御点とすべき位置）までの距離を設定します。この設定値が、パラメータ：マルチハンドベース形名で指定されたマルチハンドベース基準座標とつながって、ツール座標（メカニカルインタフェースから制御点とすべき位置までの位置関係）となります。

表 7-19：ハンドオフセット量設定パラメータ

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
マルチハンドオフセット量	MHnOFSy (n：ﾏﾙﾁﾊﾝﾄﾞ番号 1 ～ 4、 y：ﾊﾝﾄﾞ番号 1 ～ 4)	実数 3	マルチハンド番号1～4毎に、そのハンドベースに取り付けている4つのハンド(ﾌｨﾝｶﾞｰ)のオフセット量をそれぞれx、y、zで設定します。(ハンド(ﾌｨﾝｶﾞｰ)の取付位置から作業位置までの距離) 設定値はマルチハンドベース基準座標系を基準とします。単位：mm。	0.00, 0.00, 0.00



【補足】：R56TB/R57TB、又は RT ToolBox2 を使って本パラメータを設定すると、ツール座標パラメータ：MEXTLn(n：1 ～ 16) の値は自動設定されます。これにより、ツール番号を切り替えることで動作基準とするツールを選択（変更）することができます。R32TB/R33TB にはこの機能を備えていませんので、R56TB/R57TB、又は RT ToolBox2 をご使用願います。

5) ハンド初期状態の設定

接続方法の違いにより以下のパラメータを設定してください。

・RIO 接続.....Clr 命令や専用入力 (OUTRESET) などの汎用出力信号リセット時の動作を設定するパラメータ：出力信号リセットパターン (ORST700)

・パラレル I/O 接続 .....電源投入時のエアハンド I/F の出力を設定するパラメータ：ハンド初期状態 (HANDINIT)

表 7-20：ハンド初期状態設定パラメータ

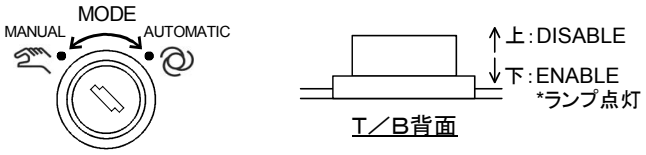
パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時設定値
出力信号リセットパターン			Clr命令や専用入力(OUTRESET)などの汎用出力信号リセット時の動作を設定します。電源ON時にもここで設定したパターンで信号出力されます。 下記のパラメータで信号ごとに32ビット単位で設定します。(OFF/ON/保持=0/1/*)	
	ORST700	文字列 4	信号番号700～731を設定します。 注)マルチハンドでは、出力信号716～723を使用します。 出力信号700～715はシステム予約領域のため設定はできません。	00000000, 00000000, 00000000, 00000000
ﾊﾝﾄﾞ初期状態	HANDINIT	整数 8	電源投入時のエアハンドI/Fの出力を設定します。 本パラメータはロボットの先端にあるハンド専用信号(900番台)の電源ON時の初期値を指定するものです。 ハンド制御を汎用IO(10000～18191番台以外)で制御する場合(HANDTYPEパラメータにて900番以外の信号を指定)の電源ON時の初期状態を設定するには、このHANDINITパラメータではなく、ORS*パラメータを使用してください。 このORS*パラメータで設定された値が電源ON時の信号の初期値になります。	1,0,1,0,1,0,1,0

(2) ハンド開閉操作 (R32TB)

以下に R32TB によるハンド開閉操作画面を示します。
マルチハンドを RIO 接続している場合のみ、画面下に表示される入出力番号 (OUT-716, IN-716) が異なるだけで、操作方法は通常のハンドの場合と同じです。ハンドは最大 6 個制御することができ、X, Y, Z, A, B, C 軸のキーそれぞれにハンド 1, 2, 3, 4, 5, 6 が割り当てられていますが、マルチハンドの場合ハンド 1 ～ 4 を使用します。ハンド開は "+" 側のキー、ハンド閉は "-" 側のキーを押すことで操作できます。

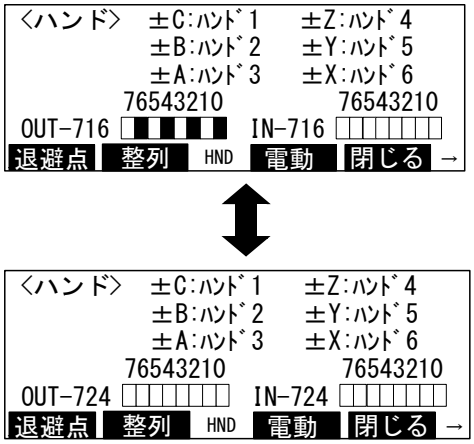
以下に操作方法を示します。

1) コントローラのモードを "MANUAL" にして、T/B の [ENABLE] スイッチを押して T/B を有効にします。(スイッチと ENABLE　LED 点灯)



2) 「HAND」キーを押して < ハンド > 画面を表示させます。
OUT-716 にはハンドの開閉状態を、IN-716 にはハンドチェック入力信号の ON/OFF 状態を示しています。
ハンド 1 を開く場合は [+X] キーを、閉じる場合は [-X] キーを押します。その他のハンドは、Y, Z, A 軸のキーで同様に操作できます。

ハンド画面の表示 [HAND]



[↓]、[↑]ｷｰにより入力信号番号を切り替えることができます。

OUT-716 ～ OUT-723	7	6	5	4	3	2	1	0
開 / 閉	閉	開	閉	開	閉	開	閉	開
ハンド番号	4		3		2		1	

IN-716 ～ IN-723	7	6	5	4	3	2	1	0
入力信号番号	723	722	721	720	719	718	717	716

注) 入力信号番号はマルチハンド製品の仕様に基づきます。

(3) ツール番号 (ツール座標パラメータ) とマルチハンドとの関係

マルチハンドを制御するため、ツール番号とツール座標パラメータを 1 ～ 16 まで拡張しています。操作の対象となるツール番号を正しく設定して、ロボットを安全にご活用願います。

⚠ 注意 プログラムの自動運転時に、ツールデータ (MEXTL1 ～ 16) を切り替えながら教示した位置へ動作させる場合は、必要な箇所で M_TOOL 変数にツール番号を代入してツールデータを切り替えて運転してください。教示時のツールデータと運転時のツール番号が一致しないと思わぬ方向へ動作しますので十分ご注意ください。

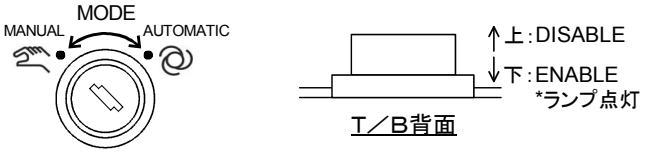
⚠ 注意 プログラムのステップ運転時に、ツールデータを切り替えて動作させる場合、教示時のツールデータとステップ運転時のツール番号が一致しないと思わぬ方向へ動作しますので十分ご注意ください。

表 7-21 : ツール番号とマルチハンドとの関係

マルチハンド番号	ハンド (ﾌｨﾝｶﾞｰ) 番号	ツール番号 (ツール座標パラメータ)
マルチハンド 1	ハンド 1	ツール 1 (MEXTL1)
	ハンド 2	ツール 2 (MEXTL2)
	ハンド 3	ツール 3 (MEXTL3)
	ハンド 4	ツール 4 (MEXTL4)
マルチハンド 2	ハンド 1	ツール 5 (MEXTL5)
	ハンド 2	ツール 6 (MEXTL6)
	ハンド 3	ツール 7 (MEXTL7)
	ハンド 4	ツール 8 (MEXTL8)
マルチハンド 3	ハンド 1	ツール 9 (MEXTL9)
	ハンド 2	ツール 10 (MEXTL10)
	ハンド 3	ツール 11 (MEXTL11)
	ハンド 4	ツール 12 (MEXTL12)
マルチハンド 4	ハンド 1	ツール 13 (MEXTL13)
	ハンド 2	ツール 14 (MEXTL14)
	ハンド 3	ツール 15 (MEXTL15)
	ハンド 4	ツール 16 (MEXTL16)

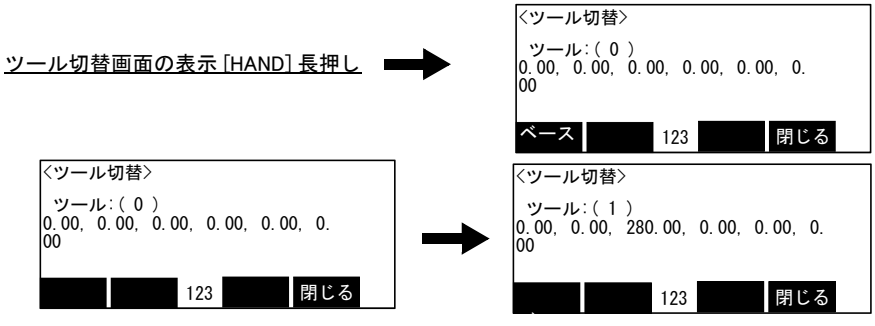
以下にツール番号の変更方法を示します。

1) コントローラのモードを "MANUAL" にして、T/B の [ENABLE] スイッチを押して T/B を有効にします。(スイッチと ENABLE　LED 点灯)



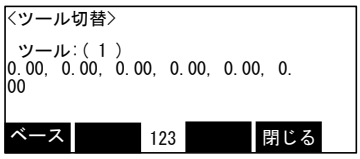
2) [HAND] キーを長押しして、< ツール切替 > 画面を表示させます。

3) 希望する数字キーを押して、[EXE] キーを押すとツールデータが切り替わります。パラメータの MEXTL1 ～ 16 は数字の 1 ～ 16 に対応しています。



ツールデータの切り替え [1] ～ [16] [EXE]

4)〝閉じる〟に割り当てられたファンクションキーを押して、終了します。



終了［F4］

5) ジョグ画面の右上に現在のツール番号(T1 ～ T16)が表示されます。

以上でツール番号の変更は終了です。

(4) アラームについて

マルチハンドに関するアラームを表 7-22 に示します。

表 7-22：エラー一覧表

エラー番号	エラー発生原因とその対策	
C190	ｴﾗｰﾒｯｾｰｼﾞ	接続中の TB は現在のツール番号に対応していません。
	原因	接続中の TB は現在のツール番号に対応していません。
	対策	現在のツールデータ(MEXTL)を確認してから TB 操作してください
H8130	ｴﾗｰﾒｯｾｰｼﾞ	リモート I/O の通信回線が異常です。
	原因	マルチハンドリモート I/O の CRC エラーもしくは接続エラーが発生しています。
	対策	通信ケーブルのチェック、または接続機器の電源を確認してください。

7.4 スプライン補間について

ロボットの動作命令の一つであるスプライン補間について説明します。

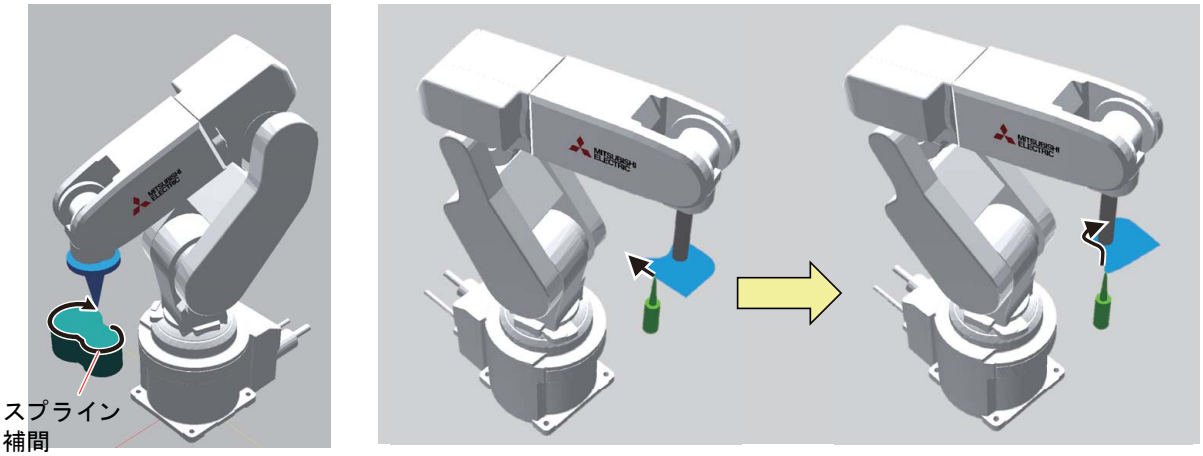
7.4.1 概要

(1) 概要

スプライン補間とは、指定された経路点を滑らかに結ぶスプライン曲線に沿って、ロボットが指定された速度で動作する機能です。

Ex-T スプライン補間とは、ロボットが把持するワーク上に指定された経路点を滑らかに結ぶスプライン曲線を、任意の座標系原点(Ex-T 座標系原点)に沿わせて、指定された速度で動作する(ワークから見た Ex-T 座標系原点が指定された速度で相対的に動作する)機能です。

シーリングや研磨･バリ取り作業などで、従来の直線・円弧補間では対応できなかった曲線状の軌跡を描く動作を可能とします。



(a) スプライン補間　　(b)Ex-T スプライン補間

図 7-10：スプライン補間の概要

(2) 特徴

・経路点として指定されたロボットの位置と姿勢を通過するように、各経路点の間に滑らかなスプライン曲線を生成します。ロボットは、その曲線に沿って線速が指定された速度となるように動作します。

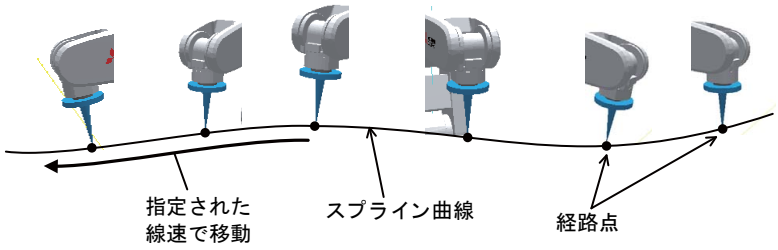


図 7-11：スプライン補間

・ロボットプログラムとは別に経路点のデータを登録したスプライン補間専用のファイル（スプラインファイル）を使用します。このファイルは RT ToolBox2 の専用の編集画面（スプラインファイル編集画面）で作成・編集し、コントローラへ書き込みます。

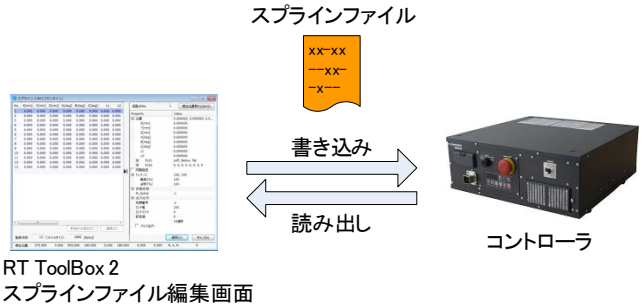


図 7-12：経路点データを専用ファイル化

・専用のファイルとして独立させたことにより、複数のロボットプログラムでスプライン補間が共用でき、さらにスプライン補間も簡単に切り替えられます。

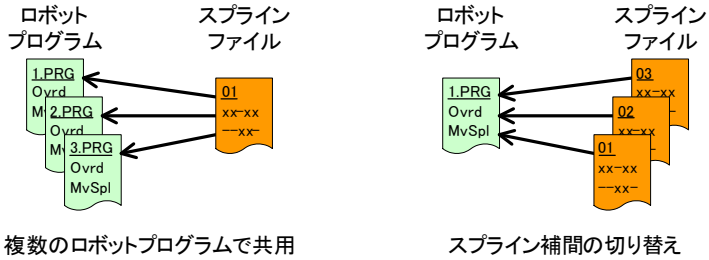


図 7-13：専用ファイル化による特徴

・スプライン曲線の膨らみを部分的に補正することや、直線・円弧軌跡に変更することができます。
・経路点のデータとして、ロボットの位置以外に信号出力や任意の数値を登録することができます。

(3) 必要な機器とソフトウェアバージョン

スプライン補間を使用するには、RT ToolBox2 が必要となります。スプライン補間は、表 7-23 に示すソフトウェアバージョンの機器で使用することができます。

表 7-23：スプライン補間が使用可能なソフトウェアバージョン

機器	機能	コントローラ	RT ToolBox2
使用可能なソフトウェアバージョン	スプライン補間	F-Q シリーズ：Ver. R5 以降 F-D シリーズ：Ver. S5 以降	Ver. 3.10L 以降
	Ex-T スプライン補間	F-Q シリーズ：Ver. R6b 以降 F-D シリーズ：Ver. S6b 以降	Ver.3.40S 以降

ティーチングボックス（R32TB/R33TB/R56TB/R57TB）については、ソフトウェアバージョンの指定はありません。

(4) 用語

スプライン補間の説明の中で使用される用語を表 7-24 に示します。

表 7-24：用語の説明

用語	説明
経路点	スプライン曲線を生成するために指定するロボットの位置データ（直交座標値）です。
ブロック	隣接する 2 つの経路点間に生成される曲線、または線分のことです。
経路	スプライン補間命令で生成される、経路点を通過するスプライン曲線のことです。
開始位置	スプライン補間の動作を開始する経路点を表します。
終了位置	スプライン補間の動作が終了する経路点を表します。
スプラインファイル	経路点、および実行に必要な設定値等が収められたファイルで、1 つのファイルが 1 つのスプライン補間に対応します。

用語	説明
スプラインファイル編集画面	RT ToolBox2 に用意されたスプラインファイルを作成・編集・保存するため専用画面です。
DXF ファイルインポート画面	RT ToolBox2 に用意された DXF ファイルからスプラインファイルを作成・保存するための専用画面です。
Ex-T 座標系原点	ロボットの外部に任意に定義された制御対象となる座標系原点のことです。

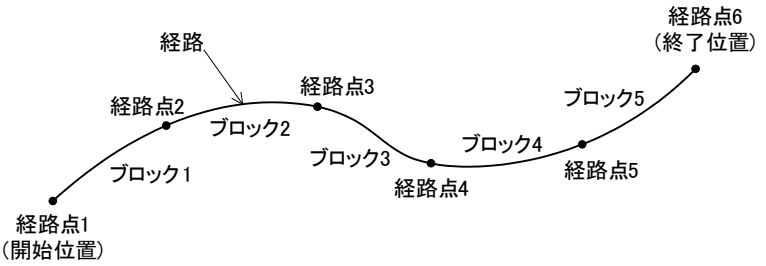


図 7-14：スプライン補間の用語

7.4.2 仕様

(1) 基本仕様

表 7-25：基本仕様

項目		仕様
対応ロボット		垂直多関節（6 軸）ロボット、水平多関節（4 軸）ロボット ※ 垂直多関節（5 軸）ロボットとユーザメカでは使用できません。
対応ロボット言語		MELFA-BASIC V にスプライン補間の命令、関数、状態変数が追加されています。 （MvSpl 命令、EMvSpl 命令、SetCalFrm 命令、SplPos 関数、SplSpd 関数、SplECord 関数、状態変数 M_SplPno、M_SplVar） ※MELFA-BASIC Ⅳ では使用できません。
動作方法		スプラインファイルをコントローラへ登録し、スプライン補間の命令を実行します。 スプラインファイルに登録された経路点を通過するようにスプライン曲線を生成し、曲線に沿ってロボットが動作します。
スプラインファイル	ファイル登録数	最大 99 本 ※ 実際に登録可能な本数は、ファイルサイズや保存領域の空き容量に依存します。
	経路点登録数	1 ファイルあたり最大 5000 点 ※5000 点で約 470KB のファイルサイズとなります。
	作成・編集・保存	RT ToolBox2 のスプラインファイル編集画面を使用して、ファイルの作成・編集・保存をおこないます。 SplWrt 命令、SplFWrt 命令を使用して、ファイルの作成・編集・保存を行います。
	経路点の登録方法	RT ToolBox2 のスプラインファイル編集画面で、以下の操作により経路点を登録します。 ティーチング操作、MDI、CSV ファイルのインポート、MXT ファイルのインポート、DXF ファイルのインポート、ロボットプログラム（SplWrt 命令）
経路補正	トレランス指定	曲線の膨らみ具合をブロック単位で指定できます。任意のブロックをスプライン曲線ではなく直線とすることもできます。
	円弧指定	連続する 3 点の経路点を指定して、スプライン曲線ではなく円弧軌跡とすることができます。 ※ スプラインファイルのバージョンによって円弧指定の方法が異なります。
	スプライン キャンセル	動作方向の変化が大きな経路点で、スプライン曲線を分割します。
	ブロック長比率	前後のブロックと比較して線分長が所定の比率以上の場合、自動的にそのブロックを直線とします。
動作モード	線速一定	指定された速度で等速に動作するように補間指令を生成します。
	線速可変	指定された速度が大きい・姿勢変化が大きい等の要因により、動作中に速度オーバーとなりそうな場合、自動的に速度を下げて動作します。
信号出力		経路点通過時に、汎用信号を出力することができます。
数値設定		経路点ごとに任意の数値を設定し、それを状態変数で参照することができます。

(2) 制限事項

表 7-26：制限事項

項目	制限の内容
付加軸	スプライン補間は付加軸を対象としていません。各経路点に付加軸の座標値が設定されていても、スプライン補間実行中は開始位置の座標値から動作しません。
Cnt 命令	Cnt 命令により補間の連続動作が指定されていても、スプライン補間の開始位置・終了位置に対しては機能しません。
Oadl 命令	Oadl 命令により最適加減速制御が有効になっていても、スプライン補間に対しては機能しません。
MvTune 命令	MvTune 命令の動作モードの指定は、スプライン補間に対しては機能しません。
FsGChg 命令	スプライン補間を FsGChg 命令（力覚制御の制御特性変更を指定）で指定する＜切り替え開始位置＞の対象とすることはできません。
ツール / ベース切替	スプライン補間の実行中断中に、ツールデータ / ベースデータを切り替えることはできません。
Jrc 命令	スプライン補間の実行中断中に、Jrc 命令を実行することはできません。
補間動作関連の状態変数	補間動作に関連する以下の状態変数は、スプライン補間の状態を返しません。 状態変数：機能 M_Acl：現在の加速時間比率を返します。 M_DAcl：現在の減速時間比率を返します。 M_Ratio：目的位置への到達率を返します。 M_RDst：目的位置までの残距離を返します。 M_Spd：現在設定されている速度を返します。
割り込み処理	割り込みの停止タイプとして「停止タイプ 1」を指定しても、スプライン補間動作中の割り込みでは「停止タイプ 2」と同じ停止となります。
トラッキング機能	トラッキング機能とスプライン補間を同時に使用することはできません。

(3) スプライン補間におけるロボットの挙動

・ロボットの現在位置とスプライン補間の開始位置が異なる場合、スプライン補間を実行すると、ロボットは開始位置まで直線補間で動作した後、スプライン補間を開始します。直線補間は、Ovrd 命令・Spd 命令・オーバーライドで指定された速度で動作します。
付加軸についても、現在位置と開始位置が異なる場合は、開始位置まで動作します。

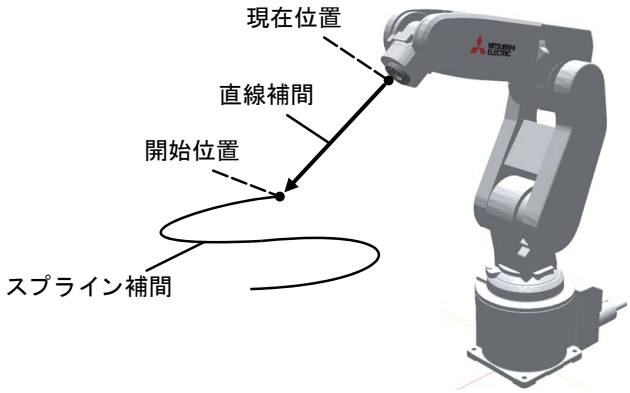


図 7-15：現在位置と開始位置が異なる場合（スプライン補間）

・ロボットの現在位置と Ex-T スプライン補間の開始位置が異なる場合、Ex-T スプライン補間を実行すると、ロボットは開始位置まで Ex-T 直線補間で動作した後、Ex-T スプライン補間を開始します。Ex-

T 直線補間は、Ovrd 命令・Spd 命令・オーバーライドで指定された速度で動作します。
付加軸についても、現在位置と開始位置が異なる場合は、開始位置まで動作します。

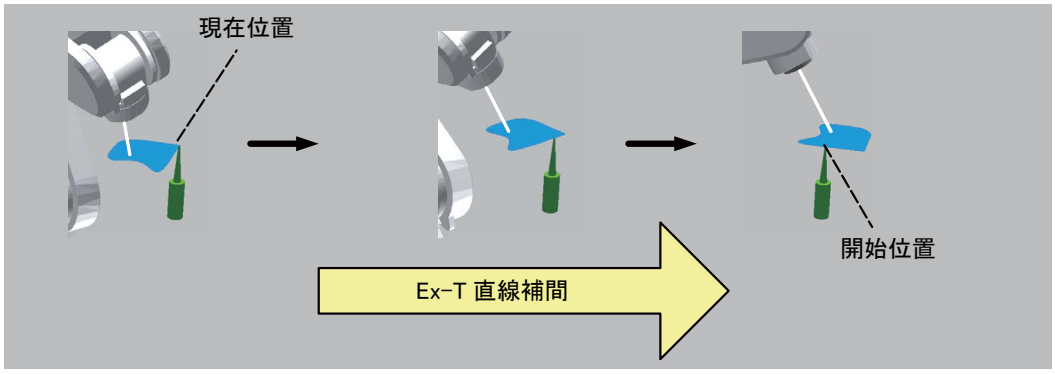


図 7-16：現在位置と開始位置が異なる場合（Ex-T スプライン補間）

・スプライン補間実行中に停止入力等により運転を中断させることができます。運転を再開させると、中断した位置から残りのスプライン補間を継続します。
・スプライン補間実行中に停止入力等により運転を中断させた後、ロボット・付加軸をジョグ操作で移動した場合、運転を再開させると、スプライン補間が中断した位置に戻ってから運転を継続します。中断位置へは関節補間で戻ります。パラメータ RETPATH（中断時ジョグ送り後の自動復帰設定）の設定は反映されません。

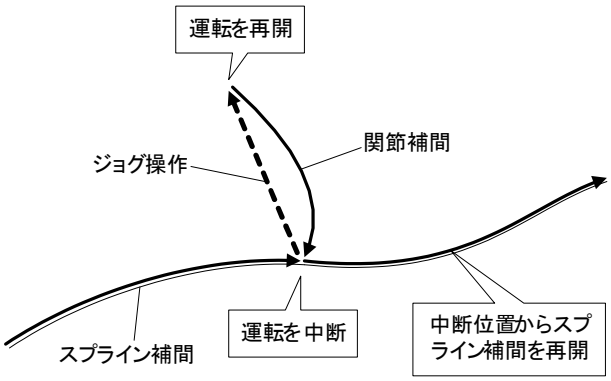


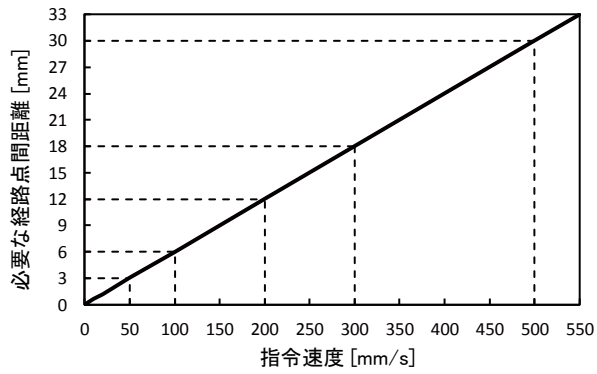
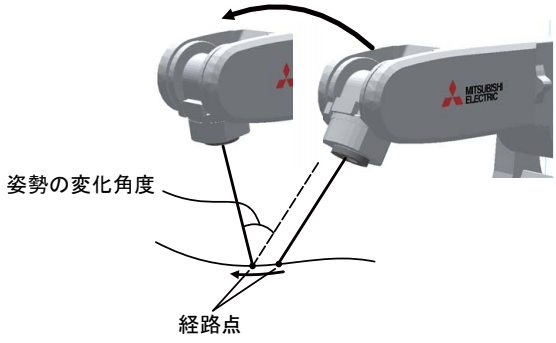
図 7-17：中断・再開時の挙動

・スプライン補間実行中に停止入力等により運転を中断させた後、実行を中断したロボットプログラムをティーチングボックス・RT ToolBox2 のプログラム編集画面に読み出すとスプライン補間の中断位置情報がリセットされます。そのため、中断行から運転を再開させると、ロボットはスプライン補間の開始位置へ移動してから運転を継続します。

(4) 経路点に関するチェック

各経路点に対して、表 7-27 に記載のチェックをおこないます。

表 7-27：経路点に関するチェック

<table>
<tr><th>種類</th><th>説明</th></tr>
<tr><td>経路点間の距離</td><td>隣接した 2 点の経路点が適切な距離を隔てているかチェックします。適切な距離は指定された速度に応じて変わり、速度に対して 2 点の経路点が近すぎる場合は、エラー L2611（経路点が近すぎます）が発生します。
下表および下図を参考に、指令速度と経路点間の距離を定めてください。
<table><tr><th>指令速度 [mm/s]</th><th>必要な経路点間の距離</th></tr><tr><td>10</td><td>0.6mm 以上</td></tr><tr><td>20</td><td>1.2mm 以上</td></tr><tr><td>50</td><td>3.0mm 以上</td></tr><tr><td>100</td><td>6.0mm 以上</td></tr><tr><td>200</td><td>12.0mm 以上</td></tr><tr><td>500</td><td>30.0mm 以上</td></tr></table>なお、この表の指令速度は、以下の式で求められます。
指令速度 ＝（MvSpl 命令の＜速度＞）×（Ovrd 命令）
 
</td></tr>
<tr><td>姿勢の変化量</td><td>隣接した 2 点の経路点の姿勢変化が大きすぎないかチェックします。姿勢の変化角度が 150 度より大きい場合は、エラー L2611（姿勢変化が大きすぎます）が発生します。
※Ex-T スプライン補間の場合、ワーク上の経路点の姿勢変化が大きすぎないかチェックします。
 
</td></tr>
<tr><td>構造フラグの値</td><td>隣接した 2 点の経路点の構造フラグの値が同じであるかチェックします。構造フラグの値が異なる場合は、エラー L2611（経路点の構造フラグが異なります）が発生します。</td></tr>
</table>

7.4.3 機能の説明

(1) 経路補正

経路補正の機能を使用することにより、スプライン曲線の形状を補正することができます。経路補正として、「トレランス指定」、「円弧指定」、「スプラインキャンセル」、「ブロック長比率」を用意しています。

■トレランス指定

スプライン曲線の膨らみ具合をブロック単位で調整したい場合、トレランス※1) を指定します。設定範囲は 0 ～ 100% で、初期状態は 100%（調整なし）です。設定値が小さいほど曲線の膨らみ具合が小さくなり、0% の場合、対象のブロックは直線となります。

姿勢に対してもトレランスを指定することができます。設定値を小さくするほど姿勢変化の膨らみ具合が小さくなりますが、経路点通過時の姿勢変化の滑らかさが減少します。

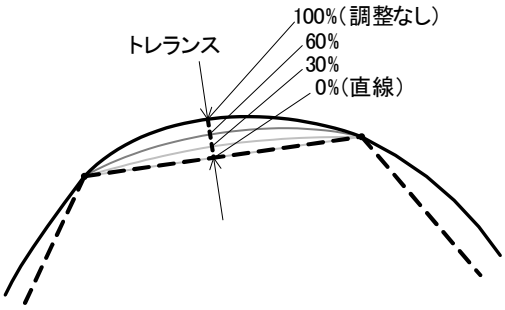
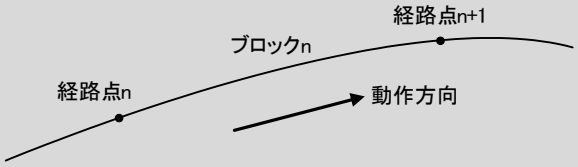


図 7-18：トレランス

◇◆◇トレランスの対象となるブロック◇◆◇

トレランスは経路点のデータとして設定します。各ブロックの開始位置側の経路点に設定されたトレランスが、そのブロックに反映されます。下図の例では、経路点 n に設定されたトレランスが、ブロック n に反映されます。



■円弧指定

スプラインファイルのバージョンによって円弧指定の方法が異なります。

1) スプラインファイルバージョン 02

連続した 2 つのブロックを円弧指定（ブロックの開始点に円弧指定）すると、その 2 つのブロックからなる曲線はスプラインではなく円弧となります。コーナー部分のように、円弧上の曲線が必要となる箇所に適用すると、スプライン曲線よりも少ない経路点数でより正確な円弧を描くことができます。

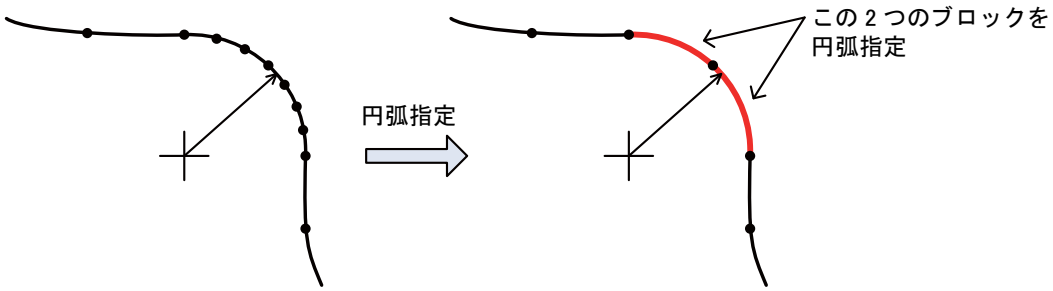


図 7-19：円弧指定（スプラインファイルバージョン 02）

※1）トレランスとは一般的に許容誤差・精度を表しますが、本機能では経路点を結ぶ直線（弦）に対するスプライン曲線の膨らみ具合を指します。

円弧指定は、必ず 2 つの連続したブロックに指定します。1 つのブロックにだけ指定してある場合は、エラー L2613（円弧の指定点数が足りません）が発生します。
また、2 つのブロックを指定することで円弧となった 3 点が直線上に並んでいて円弧が生成できない場合は、エラー L2613（ブロックデータ算出異常（Cir.Arc））が発生します。

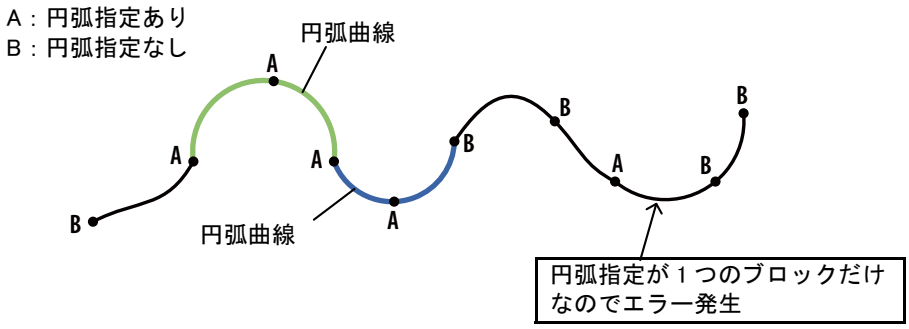


図 7-20：円弧指定でエラーとなる例（スプラインファイルバージョン 02）

2) スプラインファイルバージョン 01
連続した 3 点の経路点を円弧指定すると、その 3 点を結ぶ曲線はスプラインではなく円弧となります。コーナー部分のように、円弧状の曲線が必要となる箇所に適用すると、スプライン曲線よりも少ない経路点数でより正確な円弧を描くことができます。

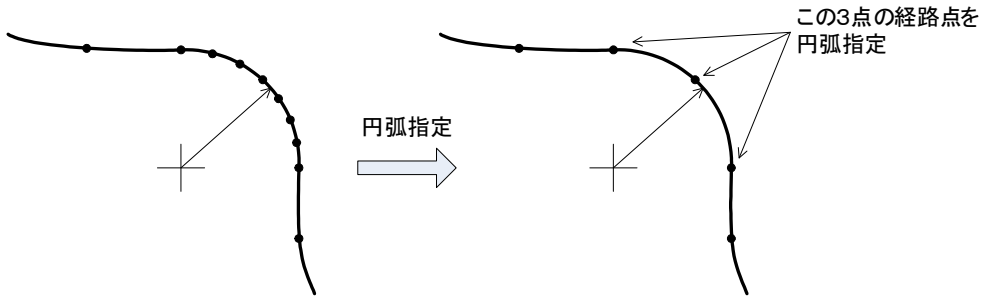


図 7-21：円弧指定（スプラインファイルバージョン 01）

円弧指定は、必ず 3 点連続した経路点に指定します。1 点のみ指定してある場合は、その指定を無視してスプライン曲線を描きますが、2 点だけ指定してある場合は、エラー L2613（円弧の指定点数が足りません）が発生します。
また、指定した 3 点が直線上に並んでいて円弧が生成できない場合は、エラー L2613（ブロックデータ算出異常（Cir.Arc））が発生します。

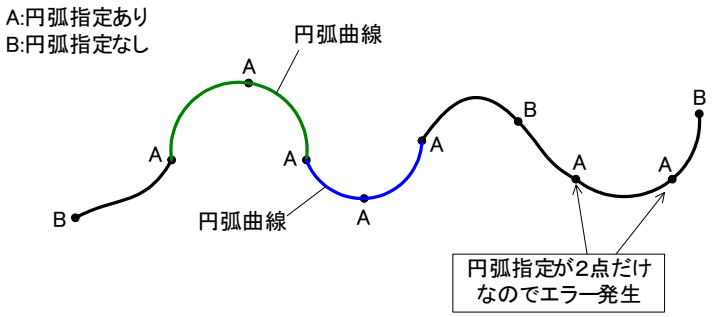
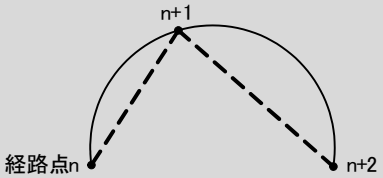


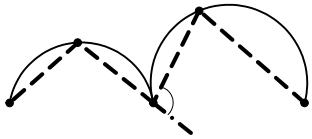
図 7-22：円弧指定でエラーとなる例（スプラインファイルバージョン 01）

◇◆◇円弧指定と他の経路補正機能との関係◇◆◇

円弧曲線を描くブロックに対して、トレランス指定・ブロック長比率は機能しません。スプラインキャンセルは、円弧の通過点（経路点 n+1）については機能しません。始点（経路点 n）と終点（経路点 n+2）については、前後のブロックのなす角がキャンセル角度より大きくなる場合、一旦減速停止します。



⚠注意　連続して円弧指定をおこなった場合、経路点の取り方によっては次図のように折り返しのきつい円弧となる可能性があります。スプラインキャンセルが働かず、円弧同士の接続点で減速停止しない場合は、急激な反転動作となりますので、生成される円弧曲線の形状には十分ご注意願います。



■スプラインキャンセル

動作方向が大きく変化する経路点をスプライン曲線で滑らかに通過するのではなく、コーナーエッジを出したい場合、キャンセル角度を指定することにより、その経路点でスプライン曲線を分割することができます。

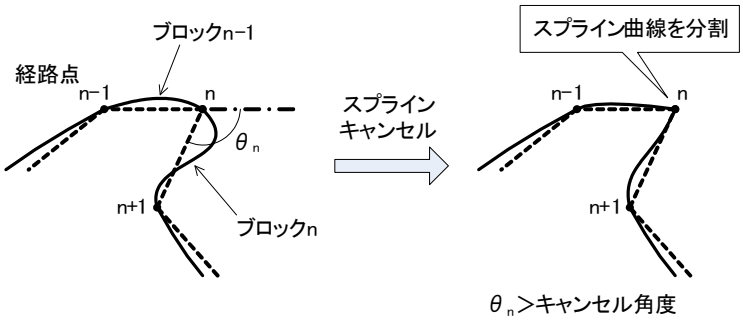


図 7-23：スプラインキャンセル

例えば、図 7-23 の経路点 n でコーナーエッジを出したい場合、キャンセル角度として経路点 n の前後のブロック（ブロック n-1 とブロック n）がなす角 θn よりも小さな角度を設定すると、経路点 n でスプライン曲線が分割されます。ロボットは、スプライン曲線を分割した経路点 n で一旦減速停止した後、再び加速し残りの補間を実行します。

キャンセル角度の設定範囲は以下のとおりで、スプライン補間ごとに個別に設定することができます。

設定範囲：0 ～ 180 度（1 度単位）

※0 度を設定した場合、スプラインキャンセルは機能しません。

※ 初期値は 120 度です。

⚠注意　スプラインキャンセルを無効とした場合やキャンセル角度を大きな値に変更した場合、動作方向が大きく変化する経路点であってもスプライン曲線は分割されず、ロボットは減速しません。そのため、ロボットの振動やサーボエラーの発生する可能性があります。

また、スプライン曲線が急激に折れ曲がるような場合は、補間位置が算出できずエラー L2163（スプライン補間指令算出異常）の発生する場合があります。

経路点の位置を確認していただき、適切なキャンセル角度を設定してください。

■ブロック長比率
　隣接するブロックに対して、ブロック長が極端に大きい場合、そのブロックをスプライン曲線ではなく自動的に直線として動作させることができます。

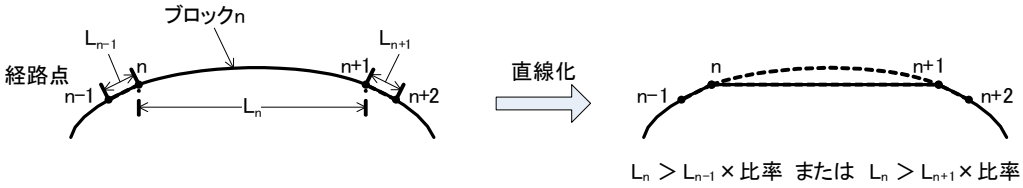


図 7-24：ブロック長比率による直線化

　ブロック ｎ の長さ L_n が、隣接するブロックの長さ L_{n-1} または L_{n+1} に対して指定の比率より長ければ、ブロック ｎ は直線となります。
ブロック長比率の設定範囲は以下のとおりで、スプライン補間ごとに個別に設定することができます。
　設定範囲：0 ～ 100 倍（1 倍単位）
　　※0 倍を指定した場合、ブロック長比率は機能しません。
　　※ 初期値は 8 倍です。
　なお、ブロック長比率によって直線となるブロックに対してトレランスが指定されていても、ブロック長比率の機能が優先されます。

◇◆◇経路補正における留意点◇◆◇
　経路補正が施された曲線と、その前後の曲線の接続部分は、補正されていないスプライン曲線同士の接続に比べて滑らかさが失われます。そのため、接続の様子や指令速度によっては、曲線の接続部分で速度が変動する場合があります。

(2) 動作モード
　スプライン補間は、指定された速度を保って制御点が移動するようにロボットを動作させます。そのため、指定された速度が大きい場合や経路点間の距離に対して姿勢の変化量が大きいような場合、動作途中で速度オーバーとなることがあります。
　そのような場合、指定された速度は保ちませんが、できるだけ速度オーバーを発生させずに動作することも選択できるよう動作モードを用意しています。表 7-28 に示すように、動作モードには 2 つのモードがあります。

表 7-28：動作モード

動作モード	説明
線速一定	指定された速度で動作し、一定の速度を保ちます。指定された速度や姿勢の変化量の大きさによっては、動作途中で速度オーバーが発生します。
線速可変	指定された速度で動作し、一定の速度を保ちます。ただし、動作途中で速度オーバーの可能性があると判断した場合は、動作速度を自動的に下げてできるだけ速度オーバーとならないように制御します。速度オーバーの可能性がなくなった場合は、元の指定速度に戻ります。 ※ 動作状況によっては、速度オーバーを抑制できない場合があります。

　動作モードは、スプライン補間ごとに個別に設定することができます。初期状態は線速一定です。
　スプライン補間の動作で速度オーバーが発生する場合、その作業が必ずしも一定の速度で動作する必要のないときは、「線速可変」をお試しください。

◇◆◇ステップ送りの場合◇◆◇
　ステップ送りでスプライン補間を実行しているときに規定の速度を超えそうになった場合は、動作モードが「線速一定」であっても動作速度を下げます。

(3) 信号出力

経路点を通過する際、任意の外部出力信号を ON/OFF することができます。この機能により、スプライン補間の動作において、例えば任意の経路点を通過したときに周辺装置へトリガ信号を出力することが、ロボットプログラムを記述しなくても実現することができます。

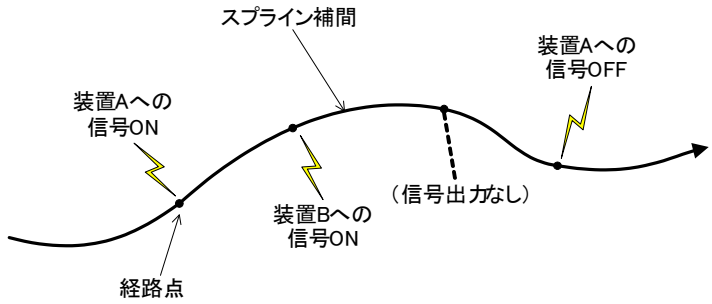


図 7-25：信号出力

信号出力をおこなうには、経路点のデータとして表 7-29 の項目を設定します。

表 7-29：信号出力の設定項目

設定項目	説明
先頭番号	出力信号の先頭番号を指定します。 設定範囲：-1 ～ 32767 ※-1 の場合、その経路点では本機能が無効となります。 初期値は -1 です。設定範囲は上記のとおりですが、実際に出力可能な先頭番号は接続されているデバイスの種類に依存します。
ビット幅	出力する信号のビット幅を指定します。 設定範囲：1, 8, 16, 32 ビット 初期値は 1 ビットです。
ビットマスク	指定されたビット幅に対して、ビット単位で信号出力を有効とするマスクパターンを指定します。 設定範囲はビット幅に応じて変わります。 ビットマスクが 1 のビットは、指定された出力データの ON/OFF 状態が信号出力に反映されます。 ビットマスクが 0 のビットは、そのときの外部出力信号の ON/OFF 状態を保持します。 初期値は 0 となっていますので、信号を出力するにはビットマスクを設定する必要があります。
出力データ	信号出力するデータを指定します。 設定範囲はビット幅に応じて変わります。 初期値は 0 です。
パルス出力	パルス出力を有効にすると、経路点を通過してから約 14ms 間だけ上記の信号出力の指定を外部出力信号に反映します。約 14ms 経過すると、外部出力信号の状態は元に戻ります。 初期値は無効です。

ビット幅	出力データ設定範囲 [16 進]
1	0, 1
8	0 ～ FF
16	0 ～ FFFF
32	0 ～ FFFFFFFF

ビット幅	出力データ設定範囲 [16 進]
1	0, 1
8	0 ～ FF
16	0 ～ FFFF
32	0 ～ FFFFFFFF

表 7-30 に経路点に設定された信号出力のデータと、外部出力信号の状態変化の例を示します。

表 7-30：信号出力の例（初期状態は全信号 OFF）

経路点	先頭番号	ビット幅	ビットマスク	出力データ	パルス出力	外部出力信号の状態 [●：ON/ ○：OFF]							
						107	106	105	104	103	102	101	100
1	100	1	1	1	無効	○	○	○	○	○	○	○	●
2	102	1	1	1	無効	○	○	○	○	○	●	○	●
3	−1	1	0	0	無効	○	○	○	○	○	●	○	●
4	100	8	5C	FA	無効	○	●	○	●	●	○	○	●
5	100	8	FF	AF	有効	●	●	●	●	●	●	●	●
	約 14ms 後 ⇒					○	●	○	●	●	○	○	●

なお、信号出力の出力先に専用出力信号が割り付けられていると、信号出力実行時にエラー L0091（この信号は専用出力に割り付け済みです）が発生します。専用出力が割り付けられたビットには信号を出力しないでください。

また、信号出力で指定した外部出力信号は占有されるわけではありません。マルチタスクにより他のロボットプログラムで対象の外部出力信号を変更すると、その変更が反映されますのでご注意願います。

◇◆◇経路点通過の判断◇◆◇

経路点の通過は、位置指令レベルで判断します。実際のロボットの位置（フィードバック位置）では判断しません。

(4) 数値設定

経路点のデータとして、任意の数値を設定することができます。

設定範囲：−1 ～ 32767（整数値）

※−1 は未設定を表します。

※ 初期値は −1 です。

状態変数 M_SplVar を参照すると、最も新しい通過済みの経路点に設定されていた値を確認することができます。複数の経路点に同一の数値を設定しておき、マルチタスクにより状態変数 M_SplVar を参照することで、スプライン補間の動作において定められた経路点を通過する際、定型の処理を実行することができるようになります。

未設定の経路点を通過したときは、状態変数 M_SplVar の値は変化しません。そのときに保持している値を返します。

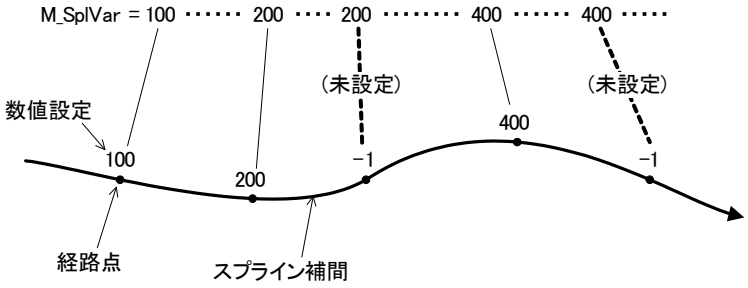


図 7-26：数値設定

(5) フレーム変換

フレーム変換とは、任意の経路に対して、その形状を保持したまま別の位置に経路を移動する機能です。

これにより、

・図面上の位置から実際の位置への変換

・作業対象の位置ずれ補正

・スプライン補間による作業エリアの変更

などに対応することが可能となります。

■処理の概略

図 7-27 を例としてフレーム変換を説明します。

任意の経路に対して基準となる座標系を設定します。座標系の設定は Fram 関数と同様に 3 つの位置データを指定します。例では、位置データ PR1, PR2, PR3 を指定して、基準座標系 ”Xfr-Zfr-Yfr” を設定しています。

次に変換後の基準座標系を同様に設定します。例では、位置データ PC1, PC2, PC3 を指定して、変換後の基準座標系 ”Xfc-Zfc-Yfc” を設定しています。

フレーム変換を実行すると、変換前と変換後で基準座標系の原点から経路点への相対的な変位が同じとなるように全経路点を算出します。例では、経路点 P11 ～ P14 から P21 ～ P24 を算出しています。

なお、変換後の各経路点の構造フラグと多回転フラグ、付加軸データには、変換前の対応する経路点と同じ値が設定されます。

また、Ex-T スプライン補間の場合は Ex-T 座標系原点 ”XE1-ZE1-YE1” もフレーム変換されます。

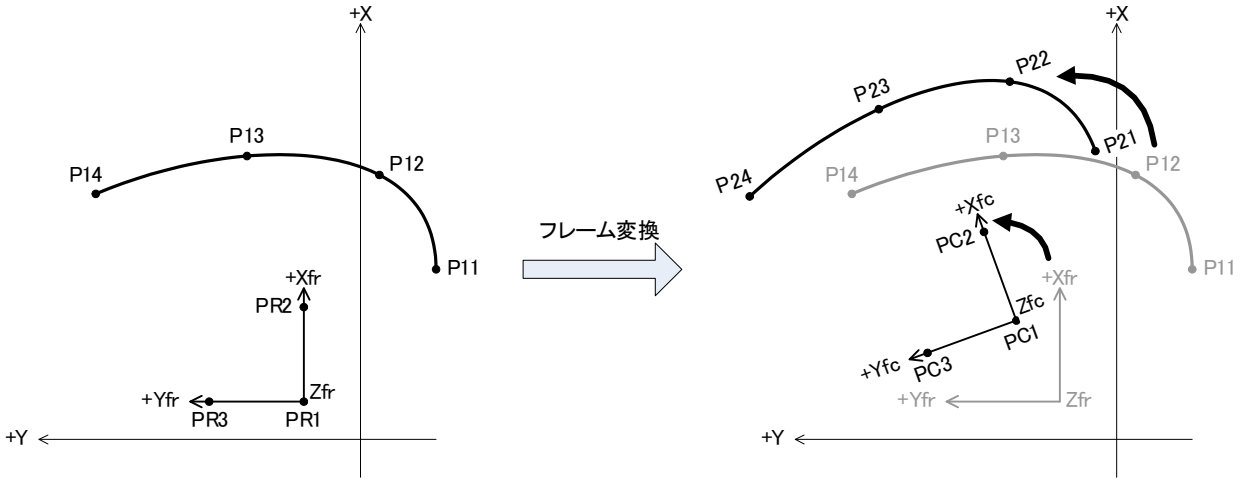


図 7-27：フレーム変換（スプライン補間）

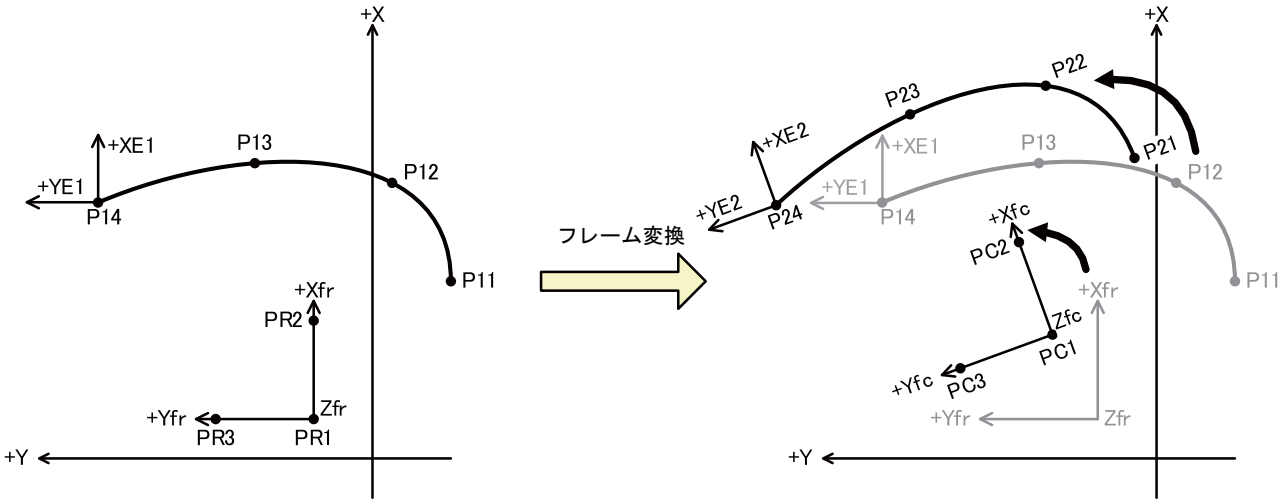


図 7-28：フレーム変換（Ex-T スプライン補間）

■設定するデータ

座標系を定義するため、Fram 関数と同様に次の 3 つの位置データを設定します。

原点位置 ..... 図 7-27 および図 7-28 の PR1, PC1 に相当

X 軸上の位置 ..... 図 7-27 および図 7-28 の PR2, PC2 に相当

XY 平面の +Y 方向の位置..... 図 7-27 および図 7-28 の PR3, PC3 に相当

変換前と変換後を合わせて、計 6 つの位置データの X,Y,Z 軸座標値を設定します。

なお、3 つの位置データの中に同一点が含まれる場合や 3 つの位置データが直線上に並んでいる場合は、座標系を定義することができず、エラー L2041（フレーム変換座標が算出できません）が発生します。

■実行方法
フレーム変換の実行方法として、3 通りの方法を用意しています。

表 7-31：フレーム変換の実行方法

実行方法	説明	使用ケース
RT ToolBox2 で実行	RT ToolBox2 のスプラインファイル編集画面でフレーム変換を実行し、変換結果を経路点としてスプラインファイルを作成します。	・変換後の経路点データをジョグ操作等によって個別に補正する場合 ・変換後の経路点データを別の経路の元データとして使用したい場合
座標系をスプラインファイルに保存	座標系の設定をスプラインファイルに保存し、スプライン補間実行時に経路点を逐次フレーム変換します。	・図面上の座標値を経路点の座標値として残しておきたい場合 ・座標系が 1 つに固定されている場合
座標系をロボットプログラムで設定	SetCalFrm 命令を使って、ロボットプログラム上で座標系を設定します。スプライン補間実行時に、設定された座標系を使って経路点を逐次フレーム変換します。	・1 つの経路に対して複数の座標系を適用したい場合 ・センサ類を使用して、経路のずれ量を毎回補正するような場合

7.4.4 作業手順
スプライン補間を実行するまでの作業手順を示します。

表 7-32：作業手順

No.	手順	作業内容
1	スプラインファイルの作成 (7.4.5 章)	RT ToolBox2 のスプラインファイル編集画面、DXF インポート機能、SplWrt 命令のいずれかを使用して、スプラインファイルを作成します。 (1) 位置の教示・インポート機能により経路点を登録します。 (2) 信号出力・数値設定を必要に応じて設定します。
2	ロボットプログラムの作成 (7.4.6 章)	スプライン補間の動作が含まれるロボットプログラムを作成します。
3	動作の確認 (7.4.7 章)	シミュレーション機能[注 1)] を使ってスプライン補間の動作を確認し、経路点の位置・ロボットプログラムを適宜修正します。
4	ロボットコントローラへ保存 (7.4.8 章)	ロボットプログラムとスプラインファイルをロボットコントローラへ書き込みます。
5	調整作業 (7.4.9 章)	RT ToolBox2 のスプラインファイル編集画面を使用して、実際のシステムに対して微調整をおこないます。 (1) 実際のシステムにおいて、デバッグ操作によりスプライン補間の動作を確認します。 (2) スプラインファイルをスプラインファイル編集画面へ読み出します。 (3) 経路点の位置・経路補正の設定を適宜調整します。 (4) スプラインファイルをコントローラへ書き込みます。 (5) 所望の経路となるよう上記 (1) ～ (4) を繰り返し、スプラインファイルのデータを調整します。
6	運転	ロボットプログラムを運転し、スプライン補間を実行します。

注 1）RT ToolBox2 mini 版では使用できません。

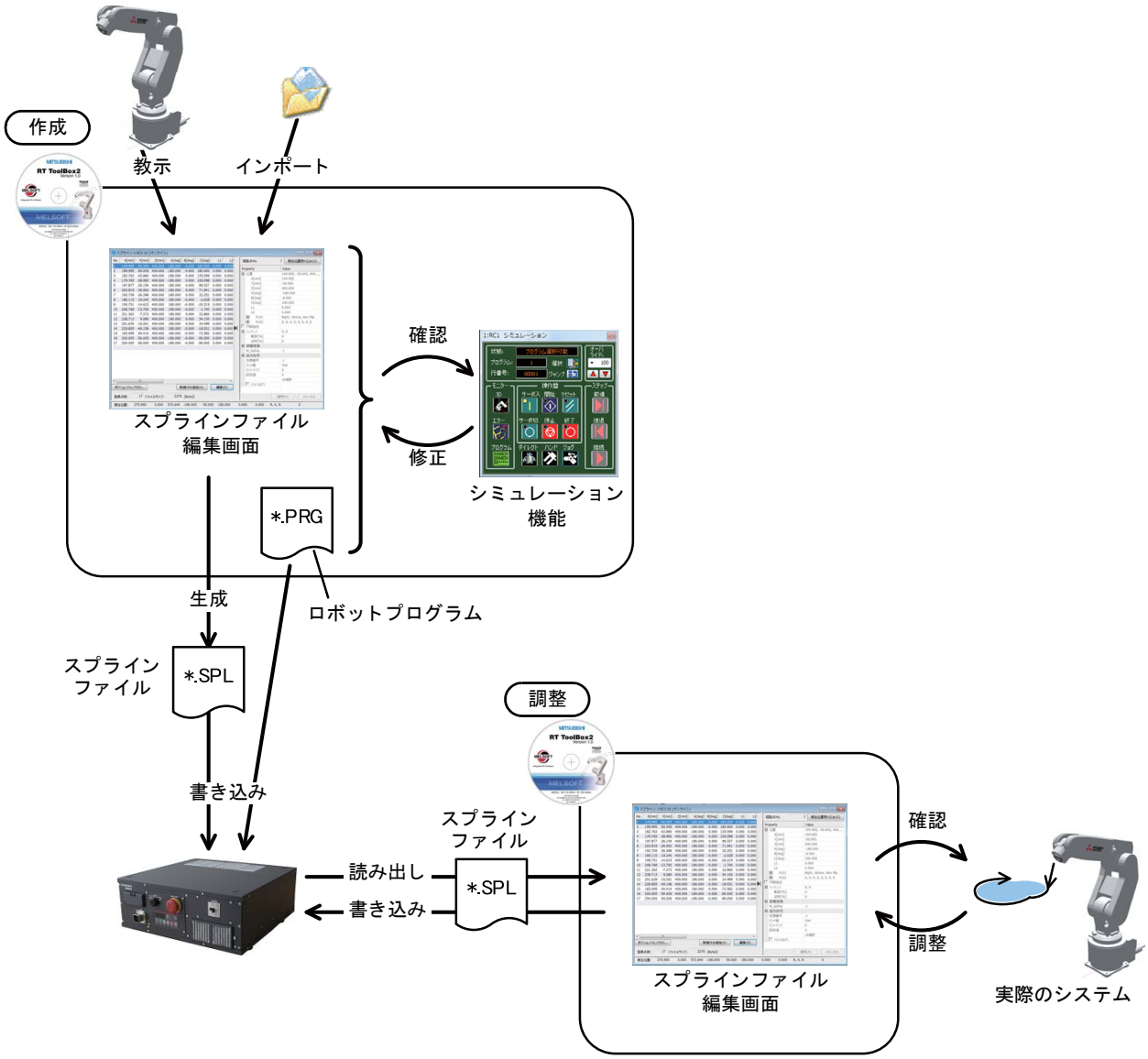


図 7-29：作業内容のイメージ

7.4.5 スプラインファイルの作成

スプラインファイルは、RT ToolBox2 のスプラインファイル編集画面、DXF インポート機能、SplWrt 命令を使用して作成します。以下に、スプラインファイルの作成方法について説明します。

(1) 新規作成

■パソコン内に新規作成

新規作成するプロジェクトの [オフライン] →[スプライン] を選択し、マウス右ボタンをクリックします。コンテキストメニューが表示されるので、[新規作成] をクリックすると、スプラインファイル編集画面が起動します。
（メニュー [ファイル] →[新規作成] をクリックしても起動します。）

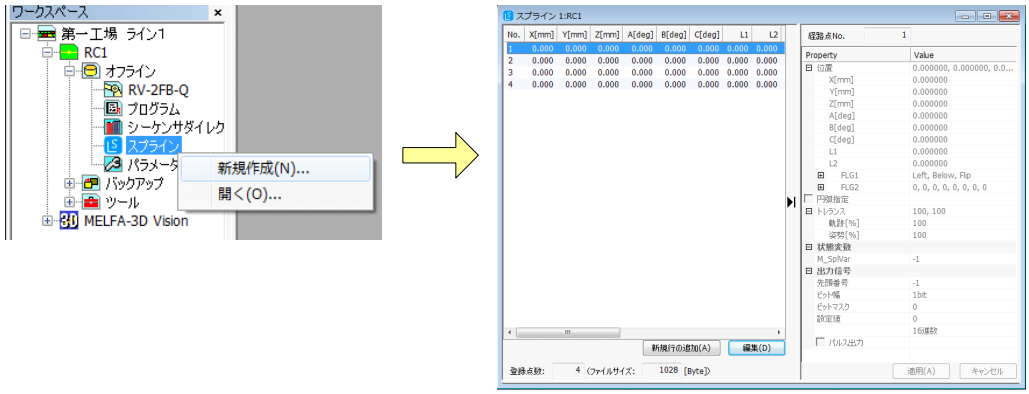


図 7-30：パソコン内に新規作成

■コントローラ内に新規作成

新規作成するプロジェクトの [オンライン] →[スプライン] を選択し、マウス右ボタンをクリックします。コンテキストメニューが表示されるので、[新規作成] をクリックすると、スプラインファイル編集画面が起動します。
（メニュー [ファイル] →[新規作成] をクリックしても起動します。）

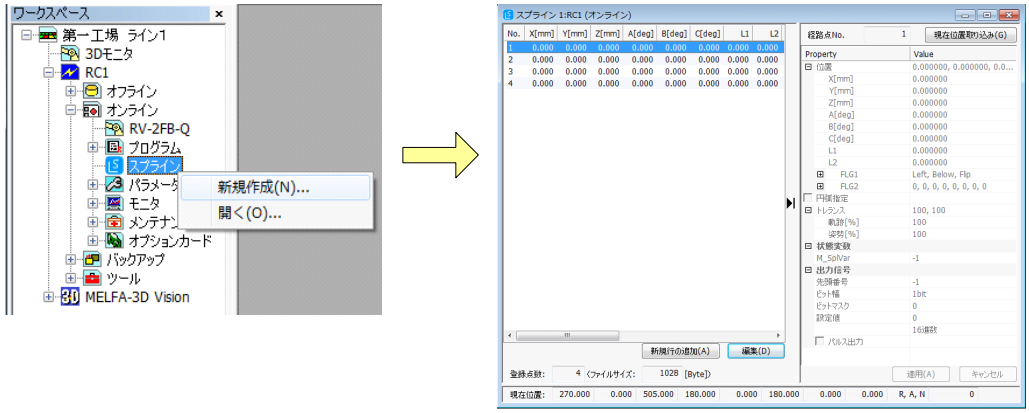


図 7-31：コントローラ内に新規作成

◇◆◇スプライン補間に対応していないロボットの場合◇◆◇

対象プロジェクトのロボットがスプライン補間に対応していない機種の場合、[オンライン] のプロジェクトツリーに [スプライン] は表示されません。[オフライン] のプロジェクトツリーには、常に [スプライン] が表示されます。

(2) DXF ファイルインポート機能で作成

この機能は RT ToolBox2 Ver.3.40S 以降で使用できます。また、DXF ファイルは AutoCAD2014 まで対応しています。

1) DXF ファイル選択

プロジェクトツリーで [ツール] → [DXF ファイルインポート] を選択して、コンテキストメニューを表示します。[開く] メニューを選択すると、DXF ファイルを開くダイアログが表示されます。

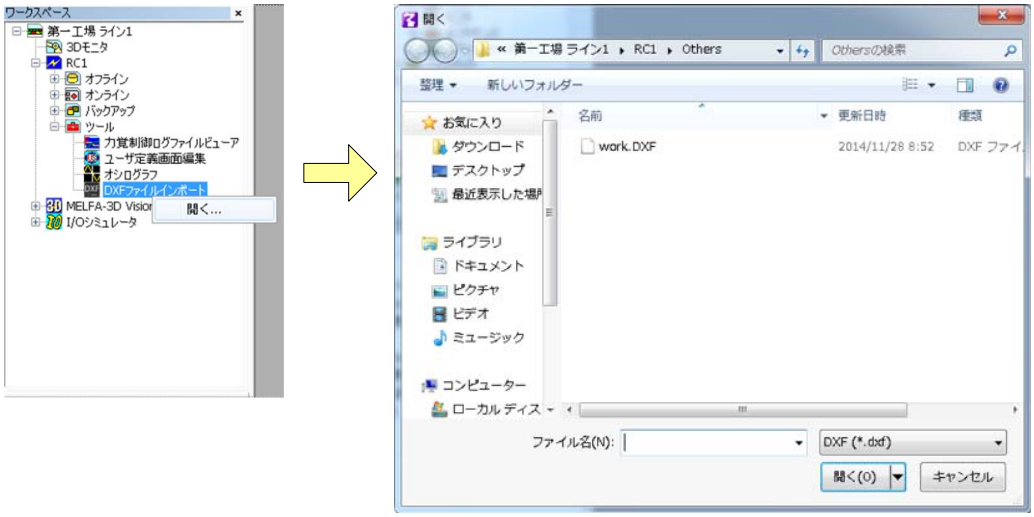


図 7-32：ファイル選択画面

選択した DXF ファイルの内容が、DXF ファイルビューアに表示されます。

※DXF ファイルでインポートできる図形データは以下のものとなっています。

- LINE (線分)
- ARC (円弧)
- CIRCLE (円)
- SPLINE (スプライン)
- POLYLINE (ポリライン)/LWPOLYLINE(ライトウェイトポリライン)

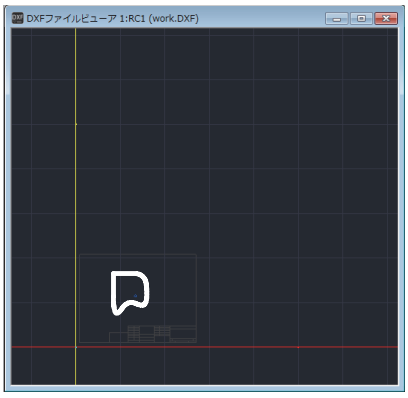


図 7-33：DXF ファイルビューア

2) 画層選択
［画層］ページの［画層一覧］リストから、インポートする画層を選択します。

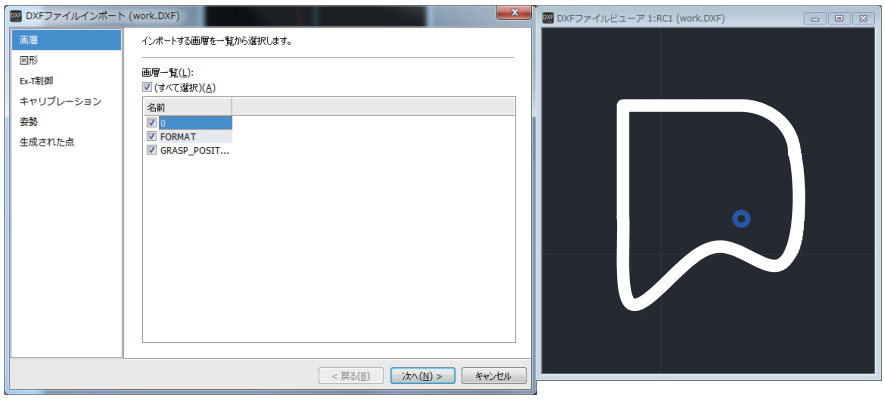


図 7-34：画層選択画面

3) 図形選択
［図形］ページで、インポートする図形を選択します。

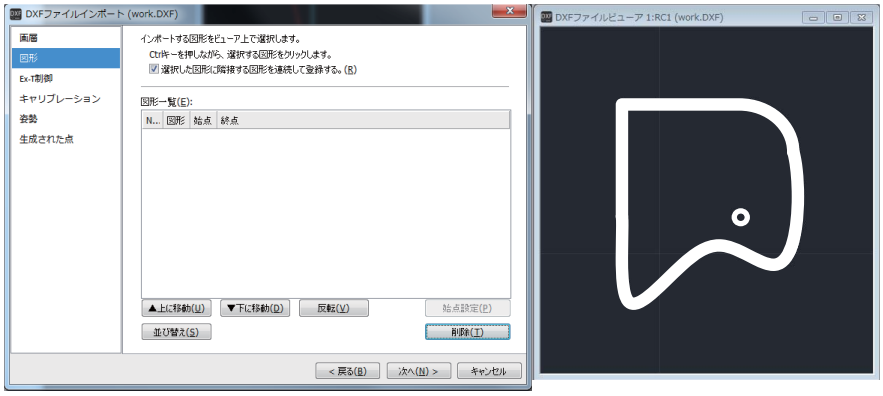


図 7-35：図形選択画面

DXF ファイルでインポートする図形を Ctrl キーを押しながらクリックすると、［図形一覧］リストに登録されます。

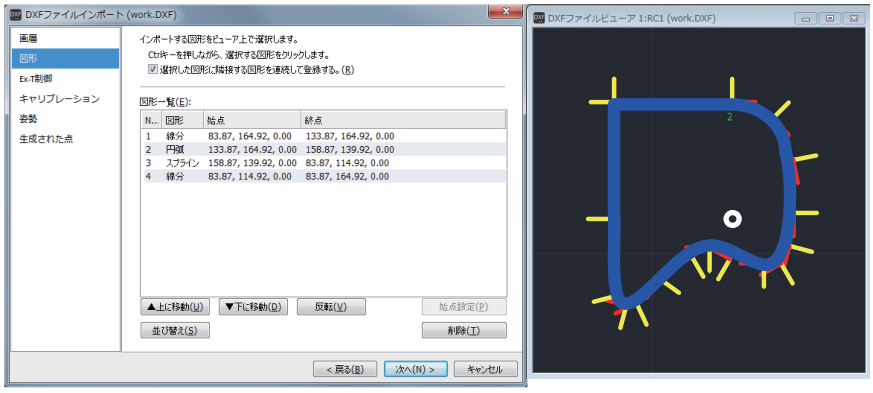


図 7-36：図形選択

先頭の図形から順に、前後の図形の始点 / 終点が連続する図形が、青色で表示されます。連続しない図形は、紫色で表示されます。

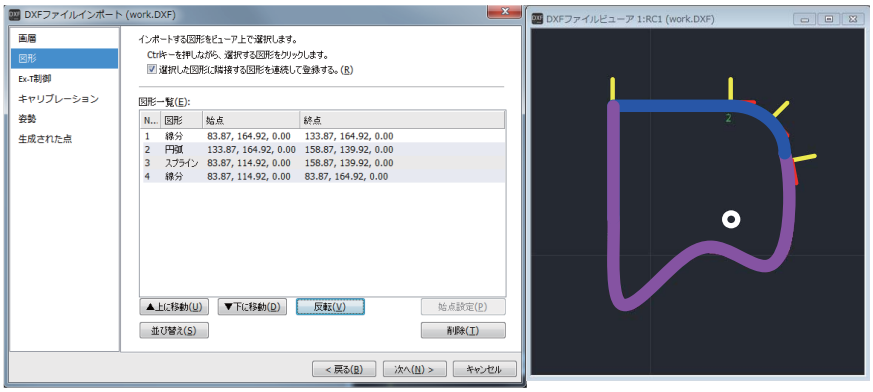


図 7-37：図形選択

[図形一覧] リストから図形を選択し、[上に移動]/[下に移動] ボタンをクリックすると、選択した図形の位置を入れ替えることができます。

図形一覧(E):

N...	図形	始点	終点
1	線分	83.87, 164.92, 0.00	133.87, 164.92, 0.00
2	円弧	133.87, 164.92, 0.00	158.87, 139.92, 0.00
3	スプライン	158.87, 139.92, 0.00	83.87, 114.92, 0.00
4	線分	83.87, 114.92, 0.00	83.87, 164.92, 0.00

下に移動

図形一覧(E):

N...	図形	始点	終点
1	円弧	133.87, 164.92, 0.00	158.87, 139.92, 0.00
2	線分	83.87, 164.92, 0.00	133.87, 164.92, 0.00
3	スプライン	158.87, 139.92, 0.00	83.87, 114.92, 0.00
4	線分	83.87, 114.92, 0.00	83.87, 164.92, 0.00

図 7-38：図形選択 (下に移動)

[図形一覧] リストから図形を選択し [反転] ボタンをクリックすると、選択した図形の始点と終点を入れ替えることができます。

図形一覧(E):

N...	図形	始点	終点
1	線分	83.87, 164.92, 0.00	133.87, 164.92, 0.00
2	円弧	133.87, 164.92, 0.00	158.87, 139.92, 0.00
3	スプライン	158.87, 139.92, 0.00	83.87, 114.92, 0.00
4	線分	83.87, 114.92, 0.00	83.87, 164.92, 0.00

反転

図形一覧(E):

N...	図形	始点	終点
1	線分	133.87, 164.92, 0.00	83.87, 164.92, 0.00
2	円弧	133.87, 164.92, 0.00	158.87, 139.92, 0.00
3	スプライン	158.87, 139.92, 0.00	83.87, 114.92, 0.00
4	線分	83.87, 114.92, 0.00	83.87, 164.92, 0.00

図 7-39：図形選択 (反転)

[並び替え] ボタンをクリックすると、確認ダイアログが表示されます。

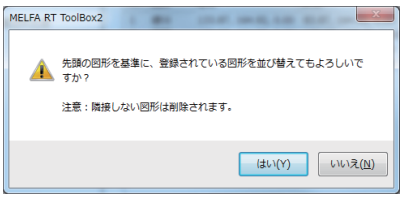


図 7-40：並び替え確認ダイアログ

確認ダイアログの [はい] ボタンをクリックすると、[図形一覧] リストに登録されている図形を先頭の図形を基準に並び替えを実行します。
※[図形一覧] リストに隣接していない図形が登録されている場合、その図形は削除されます。

図形一覧(E):

N...	図形	始点	終点
1	線分	133.87, 164.92, 0.00	83.87, 164.92, 0.00
2	円弧	133.87, 164.92, 0.00	158.87, 139.92, 0.00
3	スプライン	158.87, 139.92, 0.00	83.87, 114.92, 0.00
4	線分	83.87, 114.92, 0.00	83.87, 164.92, 0.00

並び替え

図形一覧(E):

N...	図形	始点	終点
1	線分	133.87, 164.92, 0.00	83.87, 164.92, 0.00
2	線分	83.87, 164.92, 0.00	83.87, 114.92, 0.00
3	スプライン	83.87, 114.92, 0.00	158.87, 139.92, 0.00
4	円弧	158.87, 139.92, 0.00	133.87, 164.92, 0.00

図 7-41：図形選択 (並び替え)

4) Ex-T 制御設定

[Ex-T 制御] ページで、インポートした図形の把持位置と、スプラインファイルに登録する Ex-T 座標を設定します。
ここで、Ex-T スプライン補間を使用する場合は、[Ex-T 制御を使用する。] チェックボックスをオンします。通常のスプライン補間を使用する場合は、チェックボックスをオフします。

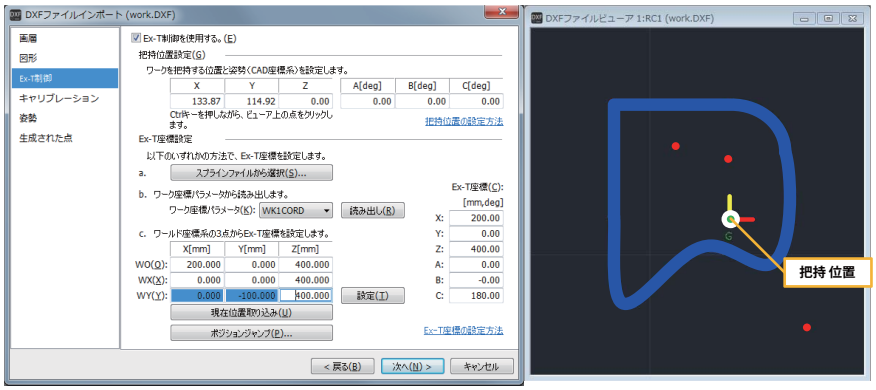


図 7-42：Ex-T 制御設定画面

Ctrl キーを押しながらビューア上の点 (赤色) をクリックすると、選択した点が把持位置として登録されます。
※ 画層 “GRASP_POSITION” の点／円（中心点）が、把持位置 (X, Y, Z) の初期値として表示されます。

Ex-T 座標の設定方法は以下の 3 種類があります。

・既存のスプラインファイルから選択
[スプラインファイルから選択] ボタンをクリックすると、スプラインファイルの選択画面が表示されます。

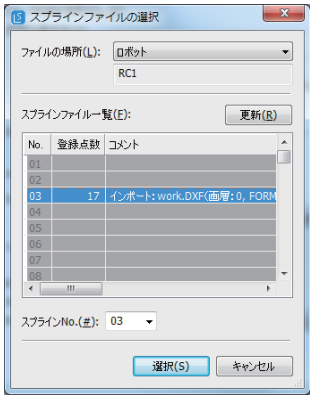


図 7-43：スプラインファイルの選択画面

選択したスプラインファイルに保存されている Ex-T 座標データを読み出します。

・ワーク座標パラメータから読出し
ドロップダウンリストから読み出すパラメータを設定し、[読み出し] ボタンをクリックすると、選択したパラメータに登録されているワーク座標データを読み出します。
・ワールド座標系の 3 点から設定
Ex-T 座標系の原点 (WO)、Ex-T 座標系の +X 軸上の位置 (WX)、Ex-T 座標系の X-Y 平面上での +Y 軸側の位置 (WY) を設定することで、Ex-T 座標データを設定します。

5) キャリブレーション

[キャリブレーション] ページで、座標系を CAD 座標系からロボットのワールド座標系に変換する

ために、キャリブレーションを実行します。
Ex-T スプライン補間を使用する場合は、キャリブレーションの必要はありません。

図 7-44：キャリブレーション画面

Ctrl キーを押しながらビューア上の点（赤色）をクリックすると、選択した点が CAD 座標系の CO/CX/CY として登録されます。
※ 画層“CALIBRATION”の点／円（中心点）が、CAD 座標系の初期値として表示します。

キャリブレーション設定には以下の 2 種類の方法があります。

・既存のスプラインファイルから選択
　[スプラインファイルから選択] ボタンをクリックすると、Ex-T 制御設定と同様のスプラインファイル選択画面が表示されます。画面で選択したスプラインファイルに保存されているキャリブレーションデータを読み出します。
・CAD 座標系、ワールド座標系の 3 点から設定
　CAD 座標系における原点、+X 軸上の位置、X-Y 平面上での +Y 軸側の位置、ワールド座標系における原点、+X 軸上の位置、X-Y 平面上での +Y 軸側の位置を設定することで、キャリブレーションデータを設定します。

6) 姿勢登録
　[姿勢] ページで、経路点の姿勢を設定します。
　姿勢タイプに [制御点に対して一定] を選択すると、各経路点の接線方向をツール方向として姿勢が登録されます。

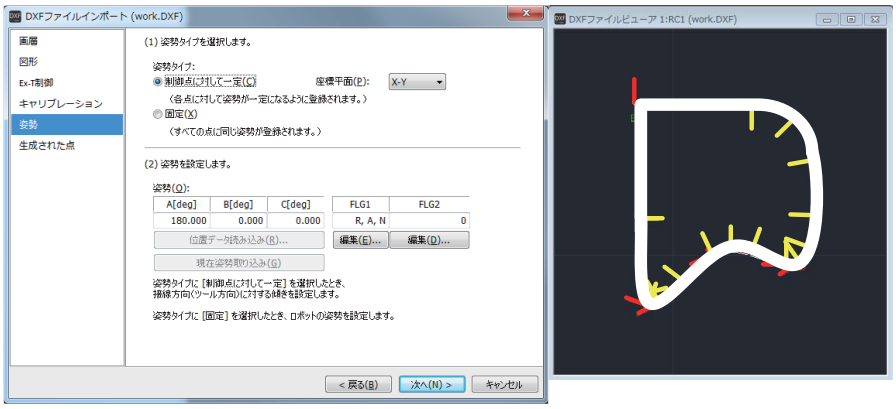


図 7-45：姿勢登録画面（制御点に対して一定）

姿勢タイプに [固定] を選択すると、全ての経路点に開始位置と同じ姿勢が登録されます。

図 7-46：姿勢登録画面 (固定)

7) スプラインファイルへの出力

[生成された点] ページで 出力形式として [スプラインファイル] を選択し、[完了] ボタンをクリックすると、スプラインファイルへと生成された点列データが出力されます。インポート完了ダイアログにて [はい] ボタンをクリックすると、スプラインファイルの保存画面が表示されます。保存するスプラインファイル番号を指定して [保存] ボタンをクリックすることでスプラインファイルを保存することができます。

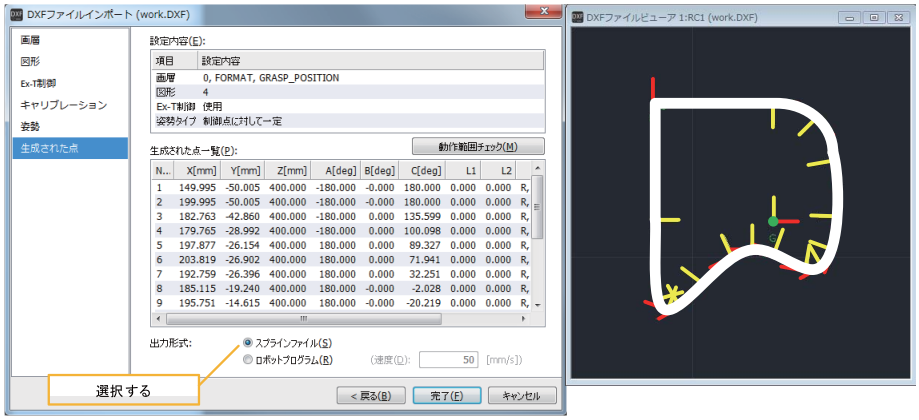


図 7-47：生成された点画面

(3) ロボット言語で新規作成

■ SplWrt 命令で新規スプラインファイルを作成

ソフトウェアバージョンが R6b または S6b 以降のロボットコントローラでは、SplWrt(スプラインライト) 命令を使用してスプラインファイルを作成することができます。命令の詳細は SplWrt (スプラインライト) を参照願います。

読み込むファイルには、経路点データとして表 7-37 に示すデータをカンマ区切りで順に記述します。また、記述例は図 7-80 となります。

1 行目には必ず「データ識別タグ」を <X> から順に記述してください。「データ識別タグ」が記述されていない場合や各行のデータの個数 (20 個) に過不足がある場合はエラーが発生します。

■ SplFWrt 命令で既存スプラインファイルを編集

ソフトウェアバージョンが R6b または S6b 以降のロボットコントローラでは、SplFWrt(スプラインフレームライト) 命令を使用して保存されているスプラインファイルにフレーム変換座標データを登録することができます。命令の詳細は SplFWrt (スプラインフレームライト) を参照願います。

(4) 既存のスプラインファイルを開く

■パソコンに保存したスプラインファイルを開く

対象となるプロジェクトの [オフライン] → [スプライン] を展開します。保存されているスプラインファイルがプロジェクトツリーに表示されるので、編集するスプラインファイルをダブルクリックします。

(スプラインファイルを選択し、マウスのコンテキストメニュー [開く] をクリックしても開きます。)

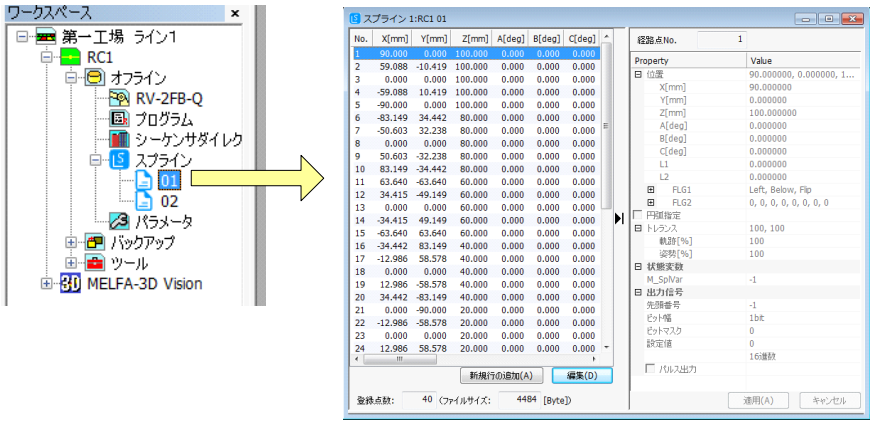


図 7-48：パソコンに保存したスプラインファイルを開く

■コントローラ内のスプラインファイルを開く
　対象となるプロジェクトの [オンライン] → [スプライン] を展開します。コントローラに保存されているスプラインファイルがプロジェクトツリーに表示されるので、編集するスプラインファイルをダブルクリックします。
（スプラインファイルを選択し、マウスのコンテキストメニュー [開く] をクリックしても開きます。）

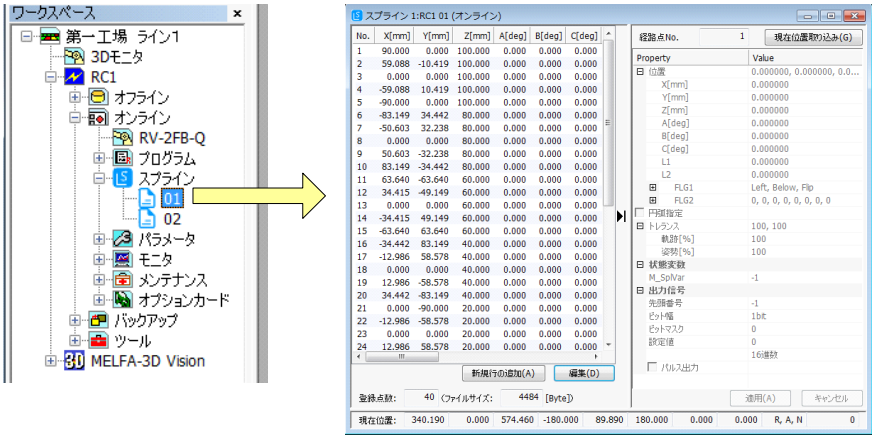


図 7-49：コントローラ内のスプラインファイルを開く

■一覧から選択してスプラインファイルを開く
　プロジェクトツリーの [スプライン] を選択し、マウス右ボタンをクリックします。
コンテキストメニューが表示されるので、[開く] をクリックすると、保存されているスプラインファイルの一覧が表示されます。
（メニュー [ファイル] → [開く] をクリックしても一覧を表示します。）

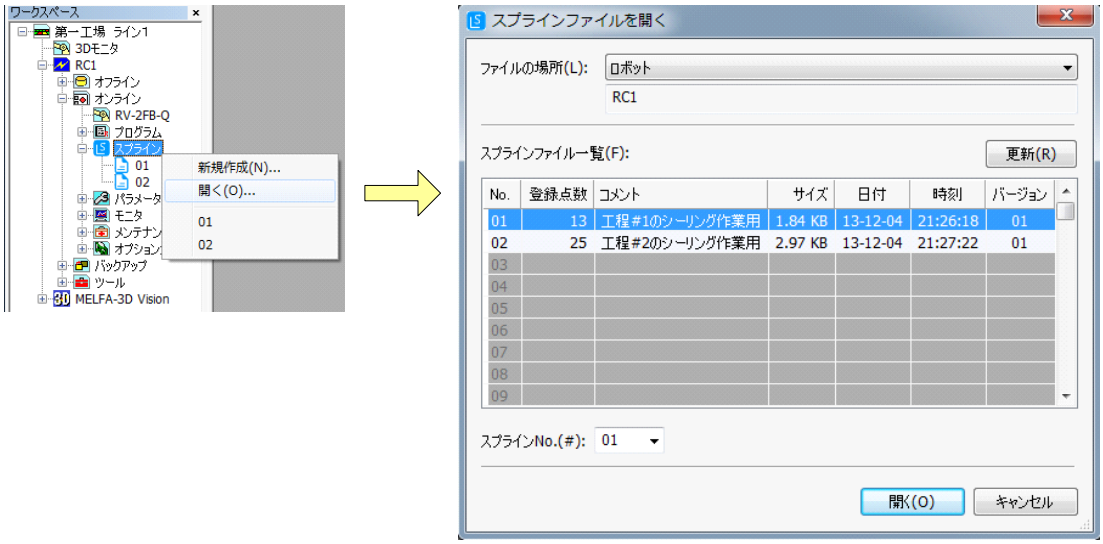


図 7-50：一覧から選択してスプラインファイルを開く

　一覧にはスプラインファイルに登録されている経路点データの数やコメントが表示されますので、それらを参考に対象のスプラインファイルを選択し、[開く] ボタンを押します。

(5) スプラインファイル編集画面の説明

スプラインファイル編集画面について説明します。

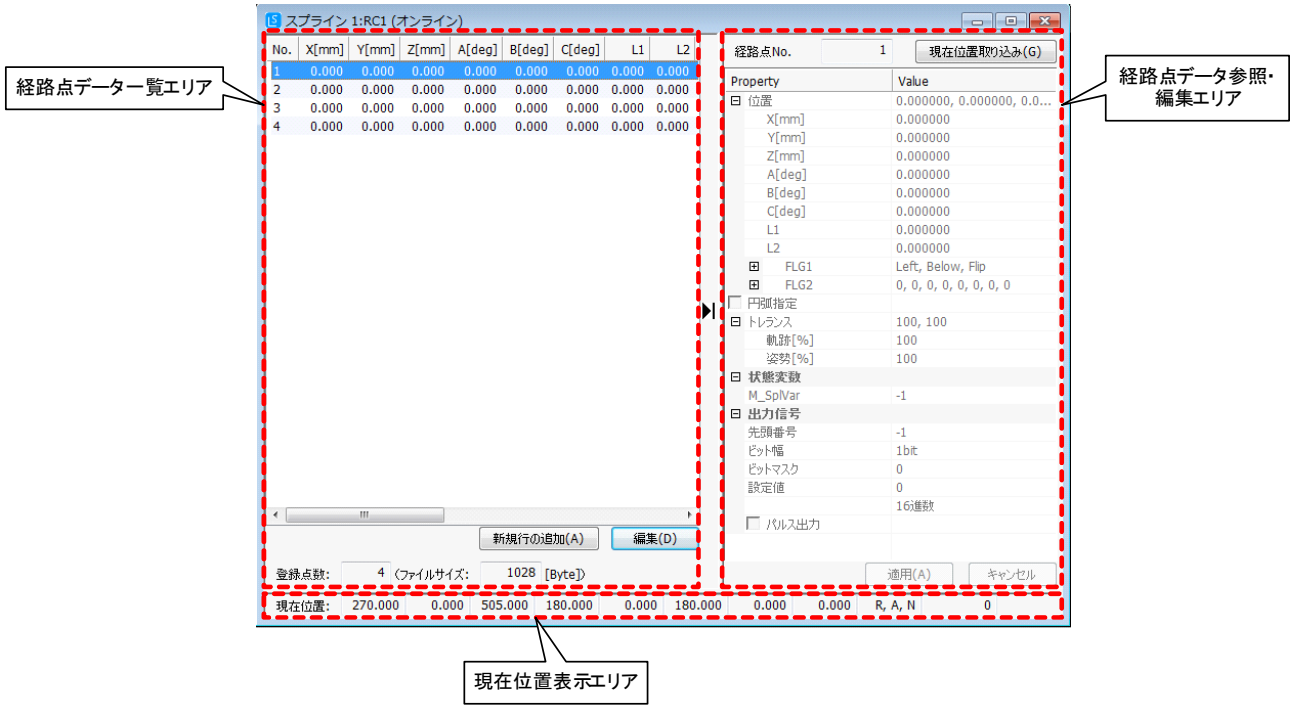


図 7-51：スプラインファイル編集画面

スプラインファイル編集画面は 3 つのエリアで構成されています。

■経路点データー覧エリア

スプラインファイルに登録されている経路点データの一覧を表示します。このエリアで以下の操作をおこないます。

・経路点データの追加・削除
・編集する経路点データの選択
・登録点数、ファイルサイズの確認

スプラインファイル編集画面の中央付近にある格納マークをクリックすると、経路点データ参照・編集エリアが格納されて経路点データー覧エリアの表示領域が広がります。これにより、経路点間のデータの変化の様子が確認しやすくなります。

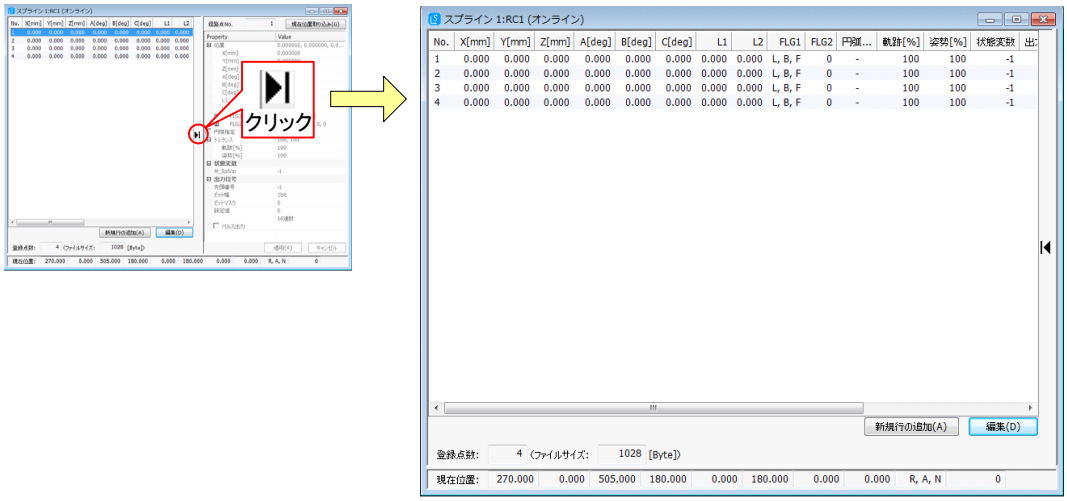


図 7-52：経路点データー覧エリアの表示領域を広げた状態

経路点データは、パソコンのキーボードとマウスの操作により、複数点を同時に選択することができます。

1点（クリック）　任意の点（[Ctrl]キー + クリック）　連続した点（[Shift]キー + クリック）　全ての点（メニュー[編集]→[編集]）

図 7-53：経路点データの選択方法

■経路点データ参照・編集エリア

経路点データー覧エリアで選択された経路点データの参照・編集をおこないます。

経路点データー覧エリアで経路点データを選択すると、その内容が経路点データ参照・編集エリアにグレーの文字色で表示されます。複数点を選択した場合は、最後に選択された経路点データが表示されます。その後、経路点データの編集を有効にすると、黒色の文字色に変わり経路点データが編集できるようになります。

（編集を有効にする方法については 617 ページの「■経路点データの編集・登録」を参照してください。）

参照状態　編集状態

図 7-54：経路点データ参照・編集エリアの参照状態と編集状態の切り替え

また、プロパティ欄の格納（[－]）・展開（[＋]）マークをクリックすると、経路点データを格納・展開して表示します。

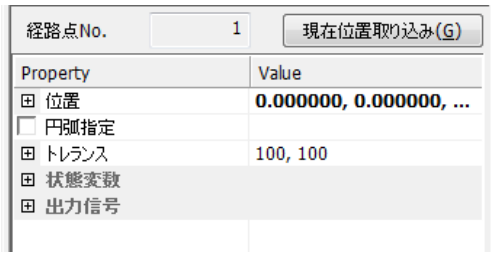


図 7-55：経路点データを格納した状態

■現在位置表示エリア

対象プロジェクトで接続しているロボットの現在位置（直交座標系）データを表示します。

対象プロジェクトがロボットと接続していない（オフラインモード）の場合、このエリアと経路点データ参照・編集エリアの [現在位置取り込み] ボタンは表示されません。

(6) スプラインファイル編集時のメニューバー

スプラインファイル編集時、メニューバーに「ファイル(F)」・「編集(E)」・「ツール(T)」が追加されます。

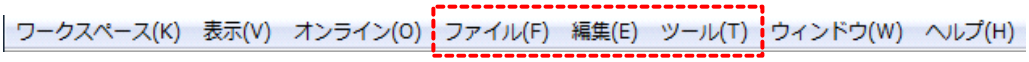
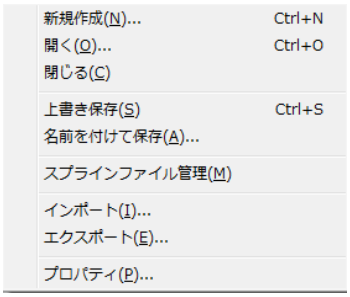


図 7-56：スプラインファイル編集時のメニューバー

それぞれのメニューの内容を以下に示します。

表 7-33：追加されるメニューの内容

メニュー項目		説明	参照ページ
ファイル  	新規作成	新たなスプラインファイルを作成するための編集画面を起動します。	602
	開く	スプラインファイルを開くための専用画面を表示します。	609
	閉じる	アクティブとなっているスプラインファイル編集画面を閉じます。	-
	上書き保存	編集中のスプラインファイルを上書き保存します。	623
	名前を付けて保存	編集中のスプラインファイルに名前を付けて保存するための専用画面を表示します。	622
	スプラインファイル管理	スプラインファイル管理画面を開きます。	626
	インポート	CSV/MXT[注 1)] 形式のファイルから経路点データを取り込みます。	633
	エクスポート	CSV 形式のファイルへ経路点データを書き出します。	636
	プロパティ	コメントの参照・記入、ファイルバージョンの変更のため、スプラインファイルのプロパティ画面を表示します。	621

メニュー項目		説明	参照ページ
編集 	元に戻す	経路点データに対して最後に適用された編集内容を取り消して、元に戻します。	637
	やり直し	[元に戻す] で取り消した内容を再び反映します。	637
	切り取り	選択した経路点データを切り取るデータとして記憶します。 [貼り付け] を実施すると、元の経路点データは切り取られます。	636
	コピー	選択した経路点データをコピーするデータとして記憶します。	636
	貼り付け	[切り取り]、[コピー] を実施した経路点データを指定した経路点データへ貼り付けます。	636
	コピー－位置データ	選択した位置データをコピーするデータとして記憶します。	637
	貼り付け－位置データ	記憶した位置データを指定した経路点データへ貼り付けます。	637
	選択行の前に挿入	選択されている経路点データの前に新たな経路点データを挿入します。	616
	コピーした経路点の挿入	[コピー] を実施した経路点データを指定した経路点データの前に挿入します。	637
	すべて選択	全ての経路点データを選択した状態にします。	－
	削除	選択されている経路点データを削除します。	618
	新規行の追加	経路点データの最後尾に新たな経路点データを追加します。	616
	編集	選択されている経路点データを編集可能にします。	617
	FLG1 一括編集	選択されている経路点の FLG1 の値を一括して編集します。	618
	FLG2 一括編集	選択されている経路点の FLG2 の値を一括して編集します。	619
	トレランス一括編集	選択されている経路点のトレランス量を一括して編集します。	619
	円弧一括指定	選択されている経路点の円弧指定の ON/OFF を切り替えます。	619
ツール 	ポジションジャンプ	ポジションジャンプ機能の専用画面を表示します。	650
	補間設定	スプラインファイルの補間設定画面を表示します。	620
	Ex-T 制御設定	スプラインファイルの Ex-T 制御設定画面を表示します。	621
	位置補正	位置補正機能の専用画面を表示します。	645
	フレーム変換	フレーム変換機能の専用画面を表示します。	647
	動作範囲チェック	編集中のスプラインファイルがロボットの動作範囲内にあるかをチェックします。 ※ この機能はシミュレーション時のみ実行可能です。	639
	経路点チェック	経路点チェック機能の専用画面を表示します。	638
	最大速度算出	編集中のスプラインファイルを使用したスプライン補間命令でエラーとならずに指定できる最大速度を算出します。	639
	スプライン曲線	編集中のスプラインファイルのスプライン曲線を表示します。	638
	ロボットプログラム－プログラム変換	編集中のスプラインファイルをロボットプログラムに変換します。	639
	ロボットプログラム－位置データ取り込み	ロボットプログラムに保存されている位置データを編集中のスプラインファイルに取り込みます。	641

注 1）MXT ファイルとは、MELFA-Works から出力された点列データが入ったファイルです。

(7) 経路点データの内容

スプラインファイルに登録する経路点データの内容を以下に示します。

表 7-34：経路点データの内容

<table>
<tr><th colspan="2">データ項目</th><th>内容</th><th>新規追加時の初期値</th></tr>
<tr><td colspan="2">位置</td><td>経路点の位置データを直交座標系（ Y, Z, A, B, C, L1, L2, FLG1, FLG2）で指定します。
距離の単位は mm、角度の単位は deg です。</td><td>X ～ L2：0.0
FLG1：Left, Below, Flip
FLG2：0</td></tr>
<tr><td colspan="2">円弧指定</td><td>円弧軌跡の経路点を指定します。
設定範囲：チェック有り（指定有効）、チェックなし（指定無効）</td><td>チェックなし</td></tr>
<tr><td rowspan="2">トレランス</td><td>位置</td><td>位置（軌跡）のトレランス量を指定します。
設定範囲：0 ～ 100 [%]</td><td>100</td></tr>
<tr><td>姿勢</td><td>姿勢のトレランス量を指定します。
設定範囲：0 ～ 100 [%]</td><td>100</td></tr>
<tr><td colspan="2">状態変数</td><td>状態変数 M_SplVar で参照する値を指定します。
設定範囲：-1 ～ 32767（-1 は未設定を表します。）</td><td>-1</td></tr>
<tr><td rowspan="4">出力信号</td><td>先頭番号</td><td>出力信号の先頭番地を指定します。
設定範囲：-1 ～ 32767（-1 は信号出力無効を表します。）</td><td>-1</td></tr>
<tr><td>ビット幅</td><td>出力する信号のビット幅を指定します。
設定範囲：1, 8, 16, 32 [ビット]</td><td>1</td></tr>
<tr><td>ビットマスク</td><td>信号出力を有効とするビットのマスクパターンを指定します。
<table>
<tr><th>ビット幅</th><th>設定範囲 [16 進]</th></tr>
<tr><td>1</td><td>0, 1</td></tr>
<tr><td>8</td><td>0 ～ FF</td></tr>
<tr><td>16</td><td>0 ～ FFFF</td></tr>
<tr><td>32</td><td>0 ～ FFFFFFFF</td></tr>
</table>
</td><td>0</td></tr>
<tr><td>設定値</td><td>信号出力するデータを指定します。
<table>
<tr><th>ビット幅</th><th>設定範囲 [16 進]</th></tr>
<tr><td>1</td><td>0, 1</td></tr>
<tr><td>8</td><td>0 ～ FF</td></tr>
<tr><td>16</td><td>0 ～ FFFF</td></tr>
<tr><td>32</td><td>0 ～ FFFFFFFF</td></tr>
</table>
</td><td>0</td></tr>
<tr><td colspan="2">パルス出力</td><td>パルス出力を指定します。
設定範囲：チェック有り（指定有効）、チェックなし（指定無効）</td><td>チェックなし</td></tr>
</table>

(8) スプラインファイルの編集

■経路点データの追加

経路点データー覧エリアの [新規行の追加] ボタンを押すと、新たな経路点データが一覧の最後尾に追加されます。
初期値は、表 7-34 の「新規追加時の初期値」に示す値となります。
（メニュー [編集] → [新規行の追加] をクリックしても追加できます。）

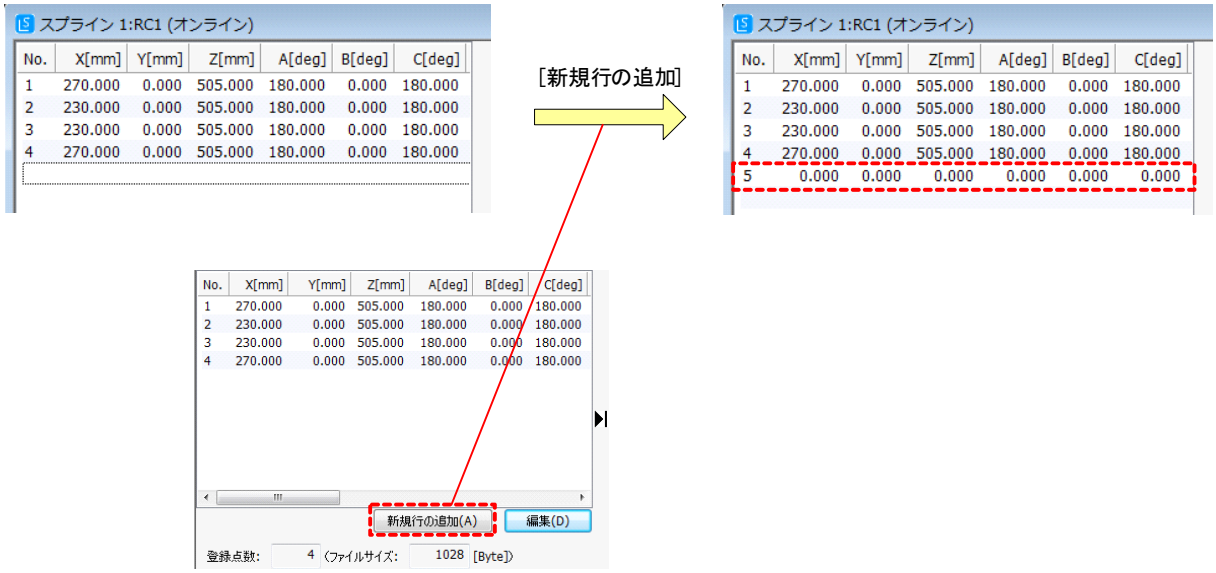


図 7-57：経路点データの追加

また、任意の経路点データを選択し、マウス右ボタンをクリックします。コンテキスト メニューが表示されるので、[選択行の前に挿入] をクリックすると、選択された経路点データの前に新たな経路点データが挿入されます。
（メニュー [編集] → [選択行の前に挿入] をクリックしても挿入できます。）

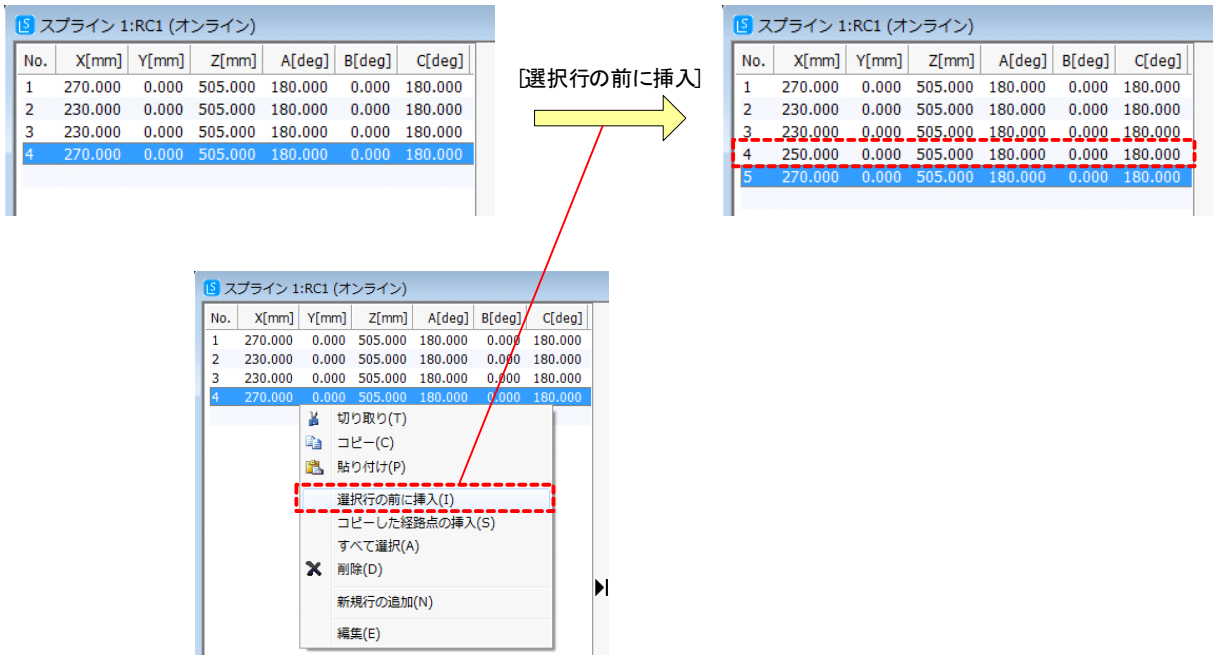


図 7-58：経路点データの挿入

このとき、選択された経路点データの行により、挿入された経路点データの初期値は表 7-35 に示す値となります。

表 7-35：挿入された経路点データの初期値

選択行	経路点データの初期値		
先頭行	表 7-34 の「新規追加時の初期値」と同じ値となります。		
先頭行以外	位置	X ～ L2	選択された経路点データと 1 つ前の行の経路点データとの平均値となります。
		FLG1, FLG2	選択された経路点データと同じ値となります。
	位置以外	表 7-34 の「新規追加時の初期値」と同じ値となります。	

■経路点データの編集・登録

経路点データを編集するには、経路点データー覧エリアで編集対象の経路点データを選択し、ダブルクリックします。ダブルクリックにより、対象の経路点データが編集有効となり、経路点データ参照・編集エリアが編集状態になります。
（経路点データー覧エリア下部の［編集］ボタン、メニュー［編集］→［編集］でも編集を有効にすることができます。）

経路点データ参照・編集エリアで編集する項目を選択し、値を設定します。

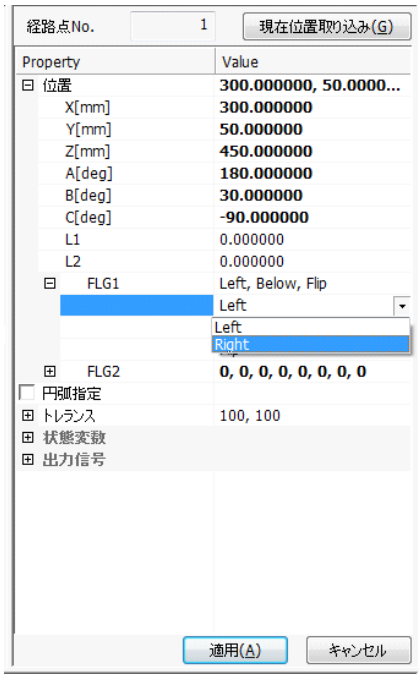


図 7-59：経路点データの編集

ロボットと接続している場合は、経路点データ参照・編集エリア上部に［現在位置取り込み］ボタンが表示されます。このボタンを押すと、ロボットの現在位置が経路点の位置データとして取り込まれます。

設定した内容を経路点データとして登録する場合は、［適用］ボタンを押します。［適用］ボタンを押さない限り、設定した内容は経路点データに登録されません。登録しない場合は、［キャンセル］ボタンを押します。［適用］・［キャンセル］ボタンを押すと、編集が無効となり、経路点データ参照・編集エリアは参照状態になります。

◇◆◇編集状態における経路点データの選択◇◆◇

経路点データ参照・編集エリアが編集状態のときは、経路点データー覧エリアで経路点データを選択することができなくなります。別の経路点データを選択する場合は、［適用］・［キャンセル］ボタンを押して、経路点データ参照・編集エリアを参照状態に切り替えてください。

■経路点データの削除
　経路点データー覧エリアで経路点データを選択し、マウス右ボタンをクリックします。コンテキストメニューが表示されるので、[削除] をクリックすると、選択された経路点データが削除されます。
（メニュー [編集] → [削除] をクリックしても削除できます。）

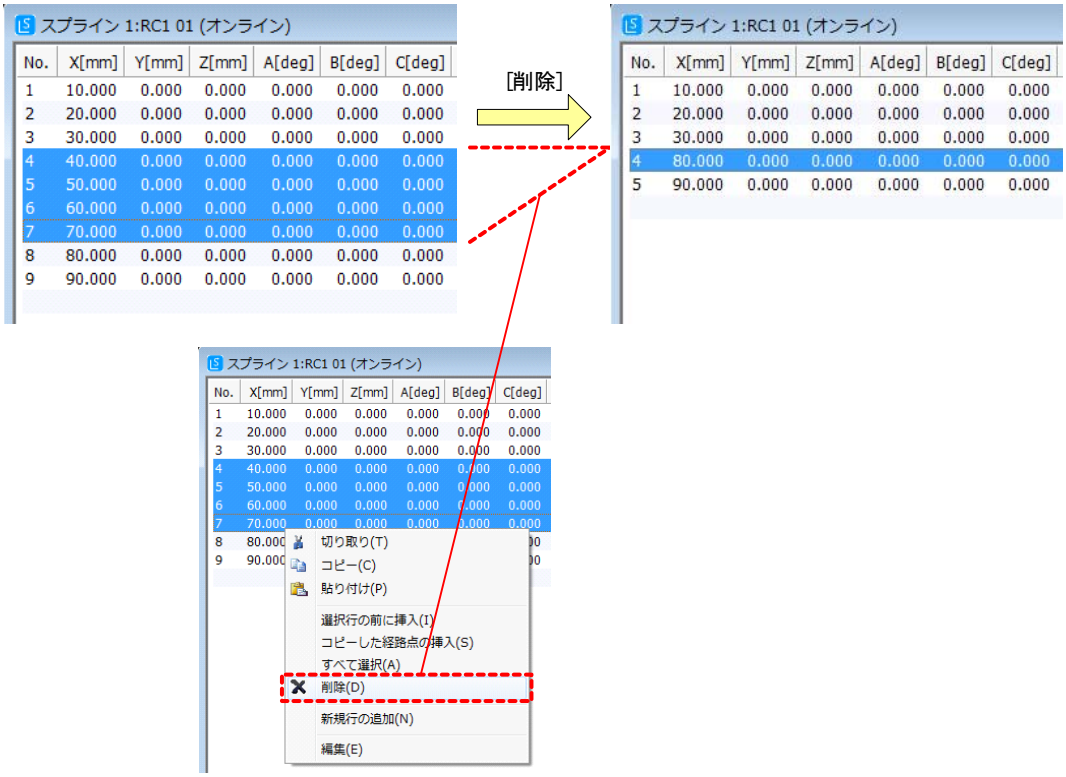


図 7-60：経路点データの削除

■ FLG1 一括編集
　選択された経路点の FLG1 の値を一括して編集します。この機能は、RT ToolBox2 Ver.3.40S 以降で使用できます。
　FLG1 の値を一括して編集するには、経路点データー覧エリアで編集対象の経路点データを選択し、マウス右ボタンをクリックします。コンテキストメニューが表示されるので、[FLG1 一括編集] をクリックすると、FLG1 一括編集画面が表示されます。
（メニュー [編集] → [FLG1 一括編集] をクリックでも表示できます。）

図 7-61：FLG1 一括編集画面

　アームの姿勢構造を変更して [OK] ボタンを押すと、変更内容が選択した経路点に反映されます。[キャンセル] ボタンを押したときは、変更内容が反映されずに FLG1 一括編集画面を閉じます。

■ FLG2 一括編集
　選択された経路点の FLG2 の値を一括して編集します。この機能は、RT ToolBox2 Ver.3.40S 以降で使用できます。
　FLG2 の値を一括して編集するには、経路点データ一覧エリアで編集対象の経路点データを選択し、マウス右ボタンをクリックします。コンテキストメニューが表示されるので、[FLG2 一括編集] をクリックすると、FLG2 一括編集画面が表示されます。
（メニュー [編集] → [FLG2 一括編集] をクリックでも表示できます。）

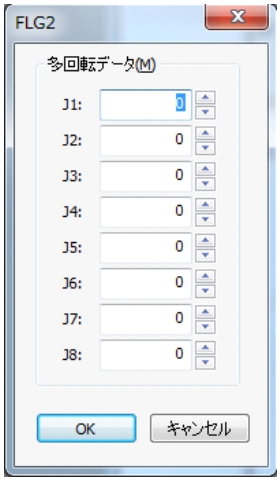


図 7-62：FLG2 一括編集画面

　各軸の多回転データを変更して [OK] ボタンを押すと、変更内容が選択した経路点に反映されます。[キャンセル] ボタンを押したときは、変更内容が反映されずに FLG2 一括編集画面を閉じます。

■トレランス一括編集
　選択された経路点のトレランス量を一括して編集します。この機能は、RT ToolBox2 Ver.3.20W 以降で使用できます。
　トレランス量を一括して編集するには、経路点データ一覧エリアで編集対象の経路点データを選択し、マウス右ボタンをクリックします。コンテキストメニューが表示されるので、[トレランス一括編集] をクリックすると、トレランス一括編集画面が表示されます。（メニュー [編集] → [トレランス一括編集] をクリックでも表示できます。）

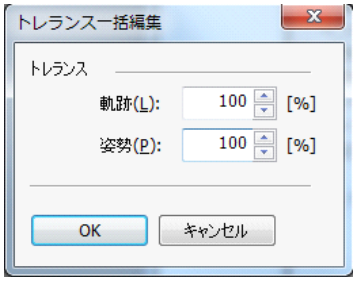


図 7-63：トレランス一括編集画面

　トレランス量を変更して [OK] ボタンを押すと、変更内容が選択した経路点に反映されます。[キャンセル] ボタンを押したときは、変更内容が反映されずにトレランス一括編集画面を閉じます。

■円弧一括指定
　選択された経路点の円弧指定を一括して編集します。この機能は、RT ToolBox2 Ver.3.40S 以降で使用できます。
　円弧指定を一括して編集するには、経路点データ一覧エリアで編集対象の経路点データを選択し、マウス右ボタンをクリックします。コンテキストメニューが表示されるので、[円弧一括指定] をクリックすると、選択した経路点の円弧指定の ON/OFF が切り替わります。（メニュー [編集] → [円弧一括指定] をクリックでも編集できます。）

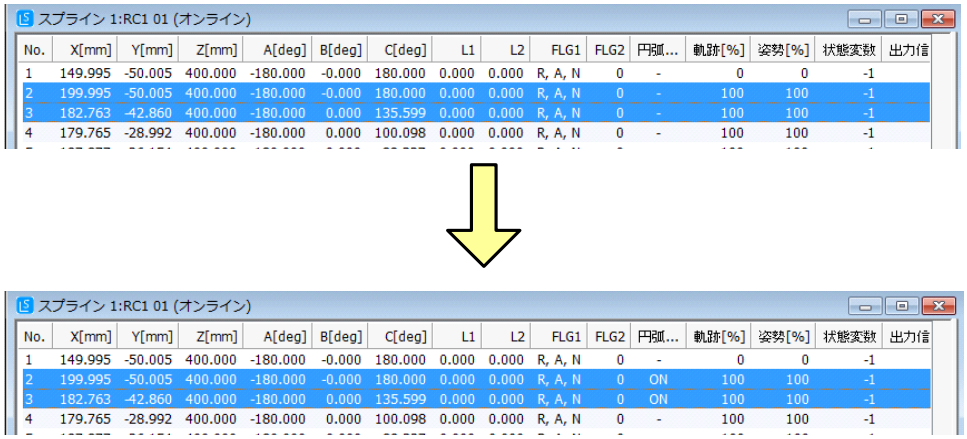


図 7-64：円弧一括指定

■補間設定

スプライン補間の「動作モード」・「キャンセル角度」・「ブロック長比率」を設定します。

メニュー [ツール] → [補間設定] をクリックすると、アクティブとなっているスプラインファイル編集画面のスプラインファイルに設定されている補間設定の内容が設定画面に表示されます。

図 7-65：補間設定画面

設定値を変更して [OK] ボタンを押すと、変更内容が記憶されて設定画面が閉じます。[キャンセル] ボタンを押したときは、変更内容を記憶せずに設定画面を閉じます。各項目の設定範囲は表 7-36 のとおりです。

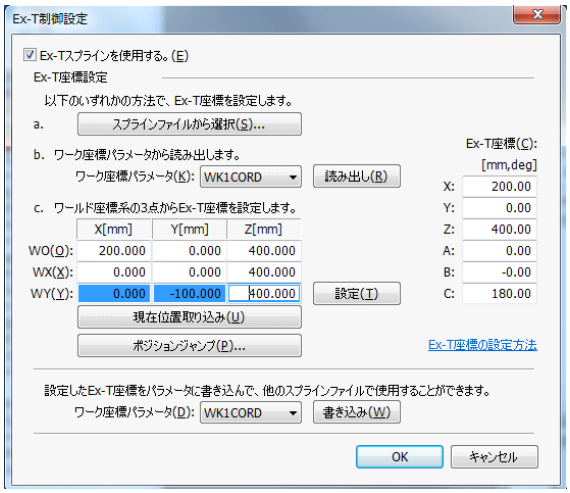
表 7-36：補間設定項目の設定範囲

項目	設定範囲	初期値
動作モード	線速一定 / 線速可変	線速一定
キャンセル角度	0 ～ 180 [度]	120
ブロック長比率	0 ～ 100 [倍]	8

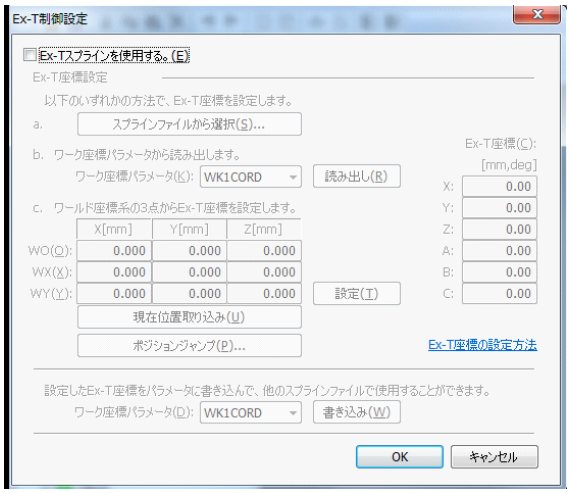
■ Ex-T 制御設定

スプライン補間の「Ex-T 制御設定」をおこないます。

メニュー [ツール] → [Ex-T 制御設定] をクリックすると、アクティブとなっているスプラインファイル編集画面のスプラインファイルに設定されている Ex-T 制御設定の内容が設定画面に表示されます。



Ex-T 制御設定あり



Ex-T 制御設定なし

図 7-66：Ex-T 制御設定画面

■コメントの記入

スプラインファイルにコメントを記入することができます。コメントに記入された内容は、スプラインファイル一覧のコメント欄に表示されます。作業内容や条件等をコメントとして記入しておくと、スプラインファイルを選択する際の一助となります。

メニュー [ファイル] → [プロパティ] をクリックすると、アクティブとなっているスプラインファイル編集画面のスプラインファイルに記入されているコメントがプロパティ画面に表示されます。

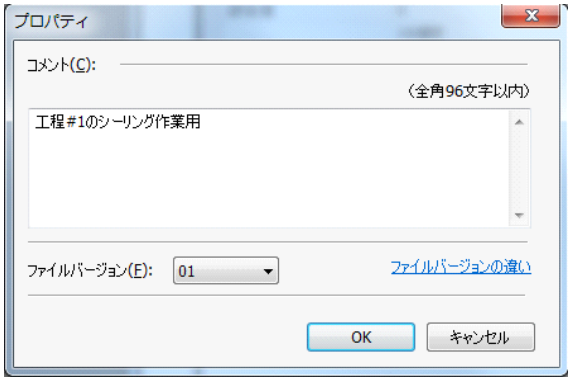


図 7-67：プロパティ画面

コメントは全角 96 文字（半角 192 文字）まで記入できます。ただし、改行は入力できません。初期状態は未記入となっています。

コメントを記入して [OK] ボタンを押すと、その内容が記憶され、プロパティ画面が閉じます。[キャンセル] ボタンを押したときは、記入したコメントを記憶せずにプロパティ画面を閉じます。

■ファイルバージョン
　スプラインファイルのバージョンを変更することができます。
　メニュー[ファイル]→[プロパティ]をクリックすると、アクティブとなっているスプラインファイル編集画面のスプラインファイルのファイルバージョンがプロパティ画面に表示されます。

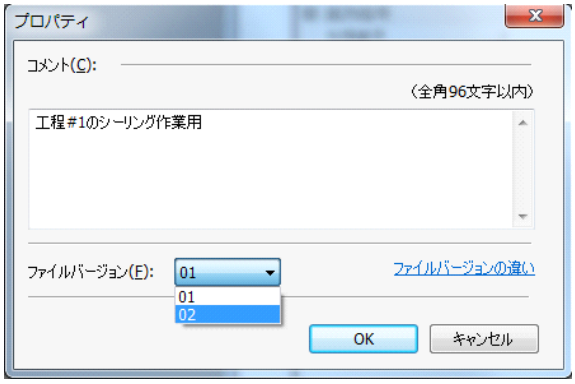


図 7-68：プロパティ画面

　ファイルバージョンを変更して[OK]ボタンを押すと、その内容が記憶され、プロパティ画面が閉じます。[キャンセル]ボタンを押したときは、ファイルバージョンの変更を記憶せずにプロパティ画面を閉じます。
　※ ファイルバージョンによる違いは 593 ページの「■円弧指定」を参照願います。

(9) スプラインファイルの保存
　スプラインファイル編集画面で編集した内容を、スプラインファイルに反映して保存します。

■メニュー[ファイル]→[名前を付けて保存]をクリック
　スプラインファイルの保存画面が表示されます。

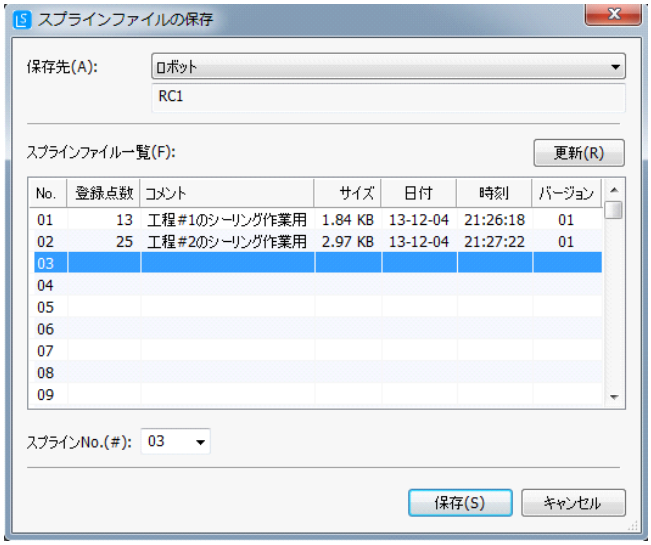


図 7-69：スプラインファイルの保存画面

　保存先を「パソコン」・「ロボット」から選択します。(オフラインモードの場合、「ロボット」は選択できません。)
　次に、スプラインファイル一覧から保存するスプライン番号を選択して[保存]ボタンを押すと、編集内容をスプラインファイルに保存して保存画面を閉じます。
　[キャンセル]ボタンを押したときは、保存をおこなわずに保存画面を閉じます。

◇◆◇スプラインファイル保存先フォルダについて◇◆◇

パソコン内のスプラインファイルは、ワークスペースのプロジェクト単位で管理されます。保存先のフォルダは、

ワークスペース作業フォルダ￥プロジェクト名￥Spline

となります。

◇◆◇スプラインファイルの名前について◇◆◇

スプラインファイルの名前は以下の形式に決められています。

SPLFILE**.SPL （大文字固定）

“**”には、01 ～ 99 のスプライン番号が入り、MvSpl 命令 /EMvSpl 命令の引数＜スプライン番号＞に対応します。

スプラインファイル編集画面では 01 ～ 99 のスプライン番号で表示しており、ファイル名を意識する必要はありません。

⚠注意　RT ToolBox2 のファイル管理画面や Microsoft Windows のエクスプローラー等でスプラインファイルのファイル名を別の名前に変更しますと、スプライン補間が実行できなくなります。
スプラインファイルの名前は変更しないでください。

■メニュー［ファイル］→［上書き保存］をクリック

既存のスプラインファイルの場合は、編集内容を対象のスプラインファイルに保存します。

新規作成のスプラインファイルの場合は、［名前を付けて保存］をクリックした場合と同じ挙動となります。

■スプラインファイル編集画面の閉じるボタンを押す

スプラインファイル編集画面右上の［×］（閉じる）ボタンを押すと、確認ダイアログが表示されます。

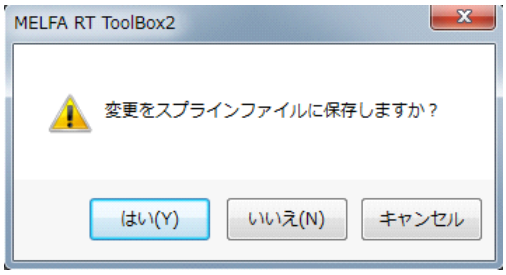


図 7-70：保存確認ダイアログ

確認ダイアログの［はい］ボタンを押すと、編集した内容をスプラインファイルに反映して保存します。

既存のスプラインファイルの場合は、編集内容を対象のスプラインファイルに上書き保存して、スプラインファイル編集画面を閉じます。

新規作成のスプラインファイルの場合は、スプラインファイルの保存画面が表示されます。

スプラインファイル一覧から保存先のスプライン番号を選択して［保存］ボタンを押すと、編集内容がスプラインファイルに保存され、スプラインファイル編集画面を閉じます。［キャンセル］ボタンを押したときは、保存をおこなわずに保存画面を閉じ、スプラインファイル編集画面に戻ります。

確認ダイアログの［いいえ］ボタンを押すと、編集した内容を破棄してスプラインファイル編集画面を閉じます。

確認ダイアログの［キャンセル］ボタンを押すと、保存をおこなわずにスプラインファイル編集画面に戻ります。

⚠注意　スプラインファイル編集画面で編集・記憶した内容は、保存操作をおこなわないとスプラインファイルに反映されません。編集作業中はこまめにスプラインファイルへの保存をおこなってください。

⚠注意　スプライン補間で使用中のスプラインファイルに対して保存することはできません。保存しようとするとエラー L2610（スプラインファイルは変更できません）が発生します。使用中のスプラインファイルに対して保存操作をおこなわないでください。

(10) スプラインファイルの削除

プロジェクトツリーに表示されている削除対象のスプラインファイルを選択し、マウス右ボタンをクリックします。
コンテキストメニューが表示されるので、[削除] をクリックすると、確認ダイアログが表示されます。

図 7-71：スプラインファイルの削除

[はい] ボタンを押すと、指定したスプラインファイルを削除します。
[いいえ] ボタンを押すと、削除操作を取りやめます。

(11) スプライン番号の変更

プロジェクトツリーに表示されている変更対象のスプラインファイルを選択し、マウス右ボタンをクリックします。
コンテキストメニューが表示されるので、[スプライン No. の変更] をクリックすると、新しいスプライン番号を指定するためのダイアログが表示されます。

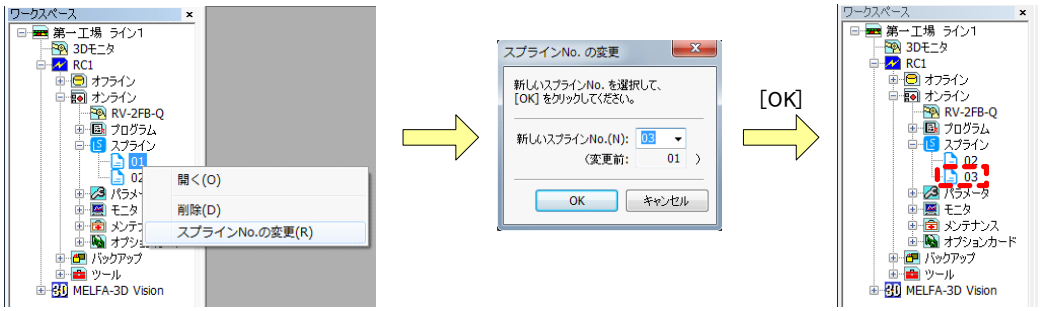


図 7-72：スプライン番号の変更

新しいスプライン番号を指定して [OK] ボタンを押すと、指定したスプラインファイルのスプライン番号が変更されます。
[キャンセル] ボタンを押すと、スプライン番号の変更操作を取りやめます。

(12) スプラインファイルのコピー

RT ToolBox2 Ver.3.20W 以降では、プロジェクトツリー上でスプラインファイルをコピーすることができます。

プロジェクトツリー上でコピー対象のスプラインファイルを選択し、コピー先のプロジェクトツリーの[スプライン]/[オンライン]/[オフライン] にドラッグ＆ドロップすると、確認ダイアログが表示されます。

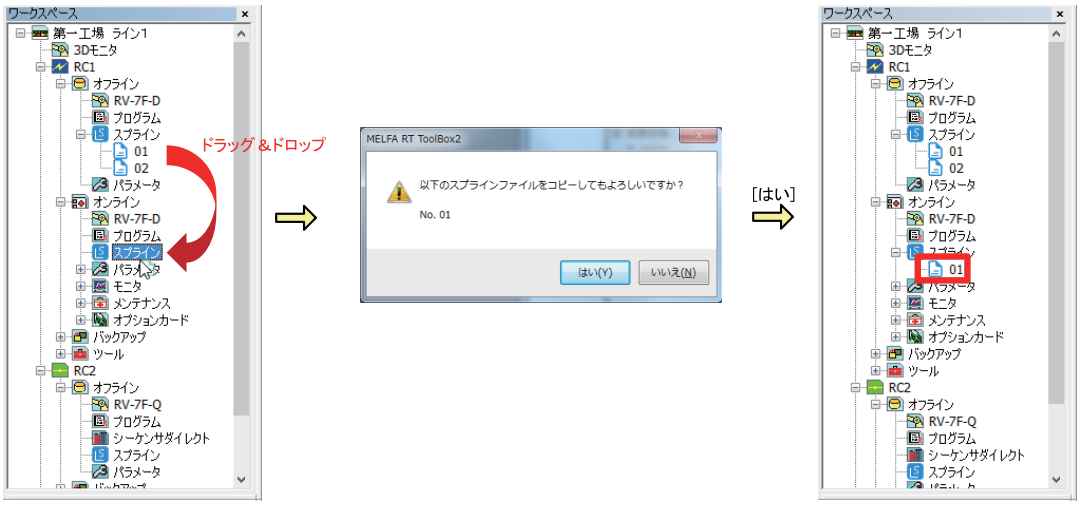


図 7-73：スプラインファイルのコピー

[はい] ボタンを押すと、指定されたスプラインファイルをコピーします。

[いいえ] ボタンを押すと、コピー操作を取りやめます。

コピー先に既に存在するスプライン番号のスプラインファイルをコピーする場合は、上書き確認ダイアログが表示されます。

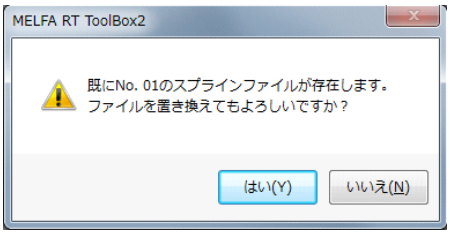


図 7-74：上書き確認ダイアログ

(13) スプラインファイル管理

RT ToolBox2 Ver.3.40S 以降ではスプラインファイルの管理画面が用意されています。

スプラインファイル管理をおこなうプロジェクトの [オンライン] → [スプライン] を選択し、マウス右ボタンをクリックします。コンテキストメニューが表示されるので、[スプラインファイル管理] をクリックすると、スプラインファイル管理画面が起動します。(メニュー [ファイル] → [スプラインファイル管理])

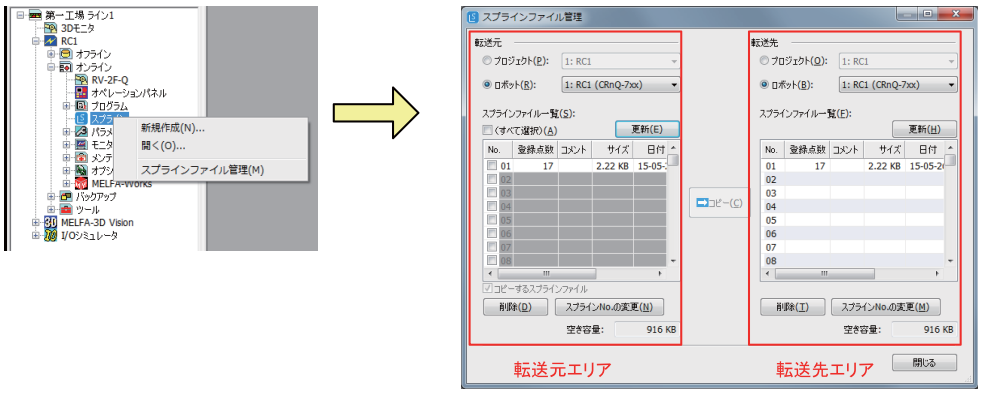


図 7-75：スプラインファイル管理画面

スプラインファイル管理画面では、2 つのエリアで構成されており、スプラインファイルのコピー、削除、スプライン No. の変更が可能となっています。

■スプラインファイルのコピー

転送元に表示されているスプラインファイルのチェックボックスにチェックを入れ、[コピー] ボタンをクリックすると、チェックを入れたスプラインファイルが転送先にコピーされます。転送先に同じ番号のスプラインファイルが存在する場合は、スプライン No. の変更をしてコピーすることができます。

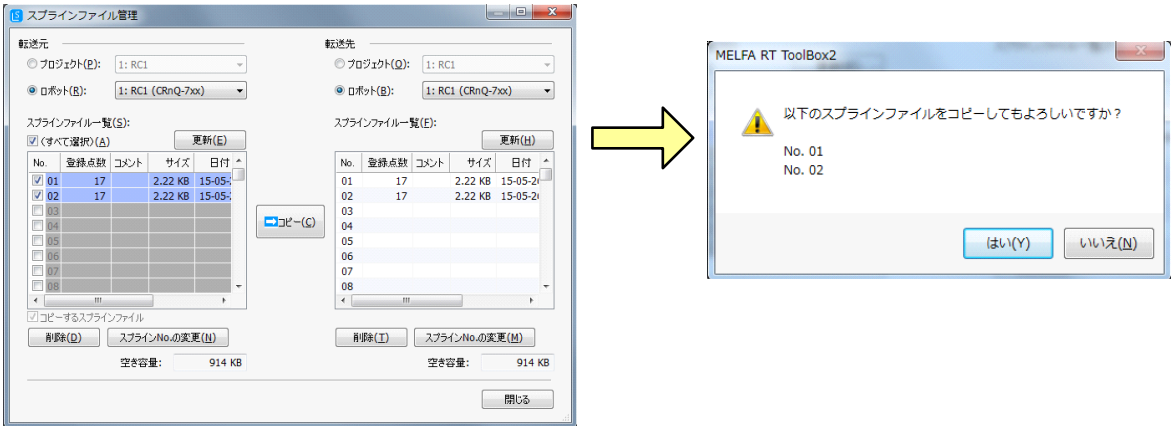


図 7-76：スプラインファイル管理画面でのファイルコピー

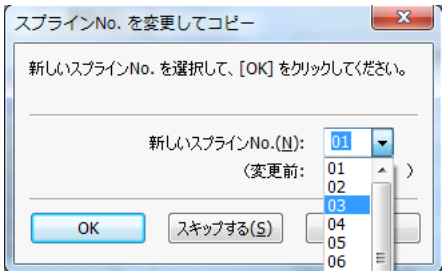


図 7-77：スプライン No. を変更してコピー

■スプラインファイルの削除
　転送元あるいは転送先のスプラインファイル一覧から削除対象のファイルを選択し、[削除] ボタンをクリックすると、選択したスプラインファイルを削除することができます。

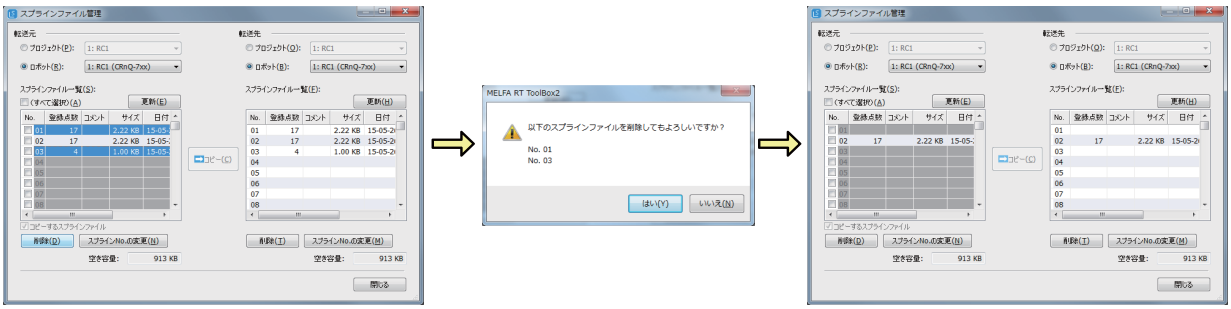


図 7-78：スプラインファイルの削除

■スプライン No. の変更
　転送元あるいは転送先のスプラインファイル一覧から変更対象のファイルを選択し、[スプライン No. の変更] ボタンをクリックすると、選択したスプラインファイルの No. を変更することができます。

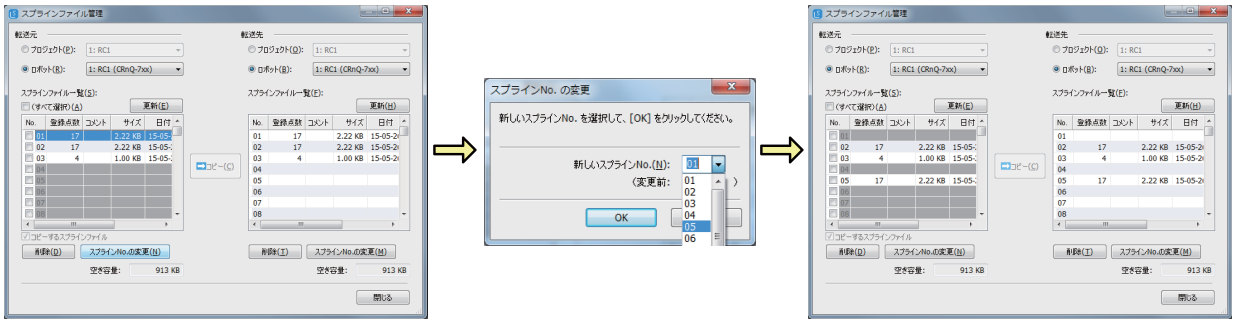


図 7-79：スプライン No. の変更

(14) インポート・エクスポート機能
　経路点データを所定の形式の CSV ファイルで記述しておくと、そのファイルをスプラインファイル編集画面へ取り込む（インポート）ことができます。これにより、例えば CAD 図面から抽出した位置データを CSV ファイルに記述しておくことにより、一括で経路点データとして登録することができます。CSV ファイルの他、MELFA-Works から出力された点列データ（MXT ファイル）も取り込むことができます。
　また、スプラインファイル編集画面から登録されている経路点データを CSV ファイルとして書き出す（エクスポート）こともできます。これにより、スプラインファイル編集画面を起動しなくても、経路点データの内容を確認することができるようになります。

■ CSV ファイルの形式
　CSV ファイルには、経路点データとして表 7-37 に示すデータをカンマ（,）区切りで順に記述します。1 行が一つの経路点データを示すように記します。

表 7-37：CSV ファイルに記述するデータ

データ識別タグ	内容
<X>	経路点の X 軸座標値を指定します。 単位：mm
<Y>	経路点の Y 軸座標値を指定します。 単位：mm
<Z>	経路点の Z 軸座標値を指定します。 単位：mm
<A>	経路点の A 軸座標値を指定します。 単位：度
<B>	経路点の B 軸座標値を指定します。 単位：度

データ識別タグ	内容
<C>	経路点の C 軸座標値を指定します。 単位：度
<L1>	経路点の L1 軸（付加軸 1）座標値を指定します。 単位：mm または度
<L2>	経路点の L2 軸（付加軸 2）座標値を指定します。 単位：mm または度
<FL1>	経路点の構造フラグの値を指定します。 設定範囲：0 ～ 7
<FL2>	経路点の多回転フラグの値を指定します。 設定範囲：0 ～ 4294967295
<Info>	円弧指定の有無を指定します。 設定範囲：指定有り（1)、指定なし（0）
<Tol1>	位置（軌跡）のトレランス量を整数で指定します。 設定範囲：0 ～ 100 [%]
<Tol2>	姿勢のトレランス量を整数で指定します。 設定範囲：0 ～ 100 [%]
<Tol3>	未使用です。100 と記述してください。
<M_SplVar>	状態変数 M_SplVar で参照する値を整数で指定します。 設定範囲：-1 ～ 32767（-1 は未設定を表します。）
<OutPort>	出力信号の先頭番地を整数で指定します。 設定範囲：-1 ～ 32767（-1 は信号出力無効を表します。）
<Bit>	出力する信号のビット幅を指定します。 設定範囲：1, 8, 16, 32 [ビット]
<Mask>	信号出力を有効とするビットのマスクパターンを 16 進数で指定します。 ビット幅 / 設定範囲 [16 進] 1 / 0, 1 8 / 0 ～ FF 16 / 0 ～ FFFF 32 / 0 ～ FFFFFFFF
<OutVal>	信号出力するデータを 16 進数で指定します。 ビット幅 / 設定範囲 [16 進] 1 / 0, 1 8 / 0 ～ FF 16 / 0 ～ FFFF 32 / 0 ～ FFFFFFFF
<Pulse>	パルス出力を指定します。 設定範囲：指定有り（1)、指定なし（0）

以下に CSV ファイルの例を示します。

```
<X>,<Y>,<Z>,<A>,<B>,<C>,<L1>,<L2>,<FL1>,<FL2>,<Info>,<Tol1>,<Tol2>,<Tol3>,<M_SplVar>,<OutPort>,<Bit>,<Mask>,<OutVal>,<Pulse>
 300.00,  125.00,  325.00,  180.00,    0.00,  180.00,  0.00,  0.00,  7,  0,  0,  100,  100,  100,     0,  10100,  8,    a,  ff,  0
 250.00,  100.00,  325.00,  180.00,    0.00,  180.00,  0.00,  0.00,  7,  0,  0,  100,  100,  100,     1,     -1,  1,    0,   0,  0
 250.00,    0.00,  325.00,  180.00,   15.00,  180.00,  0.00,  0.00,  7,  0,  1,  100,  100,  100,    10,  10100,  8,    5,  ff,  1
 300.00,  -50.00,  325.00,  165.00,    0.00,  180.00,  0.00,  0.00,  7,  0,  1,  100,  100,  100,   100,     -1,  1,    0,   0,  0
 350.00,    0.00,  325.00,  180.00,-15.00,  180.00,  0.00,  0.00,  7,  0,  1,  100,  100,  100,    -1,     -1,  1,    0,   0,  0
 300.00,   50.00,  325.00,-165.00,    0.00,  180.00,  0.00,  0.00,  7,  0,  1,  100,  100,  100,    -1,     -1,  1,    0,   0,  0
 250.00,    0.00,  325.00,  180.00,   15.00,  180.00,  0.00,  0.00,  7,  0,  1,  100,  100,  100,     0,  10100,  8,   ff,   0,  0
 250.00,-100.00,  325.00,  180.00,    0.00,  180.00,  0.00,  0.00,  7,  0,  0,   50,  100,  100,    -1,     -1,  1,    0,   0,  0
 300.00,-125.00,  325.00,  180.00,    0.00,  180.00,  0.00,  0.00,  7,  0,  0,   50,  100,  100,    -1,     -1,  1,    0,   0,  0
```

図 7-80：CSV ファイルの例

1 行目には必ず表 7-37 の「データ識別タグ」を <X> から順に記述してください。「データ識別タグ」が記述されていない場合や各行のデータの個数（20 個）に過不足がある場合はインポートできません。

■ MXT ファイルの形式

MXT ファイルとは、MELFA-Works から出力された点列データが入ったファイルです。
MXT ファイルの作成は下記の手順で行います。
※MELFA-Works の使用方法については「MELFA-Works 取扱説明書 (BFP-A8477)」を参照願います。
※ スプラインと Ex-T スプラインによって手順が異なる箇所がありますので、注意してください。

1) MELFA-Works の起動
MELFA-Works を起動します。

2) ワークの把持
※ この作業は Ex-T スプラインの場合のみ必要となります。スプラインの場合は「4) 経路作成」の作業を実施してください。
ロボットに把持用のハンドに取り付けます。
[ロボット設定] → [変更] をクリックし、ロボット詳細設定画面を表示します。
ロボット詳細設定画面の [信号接続] ボタンをクリックし、Hand I/O 画面を表示し、ワークを把持します。

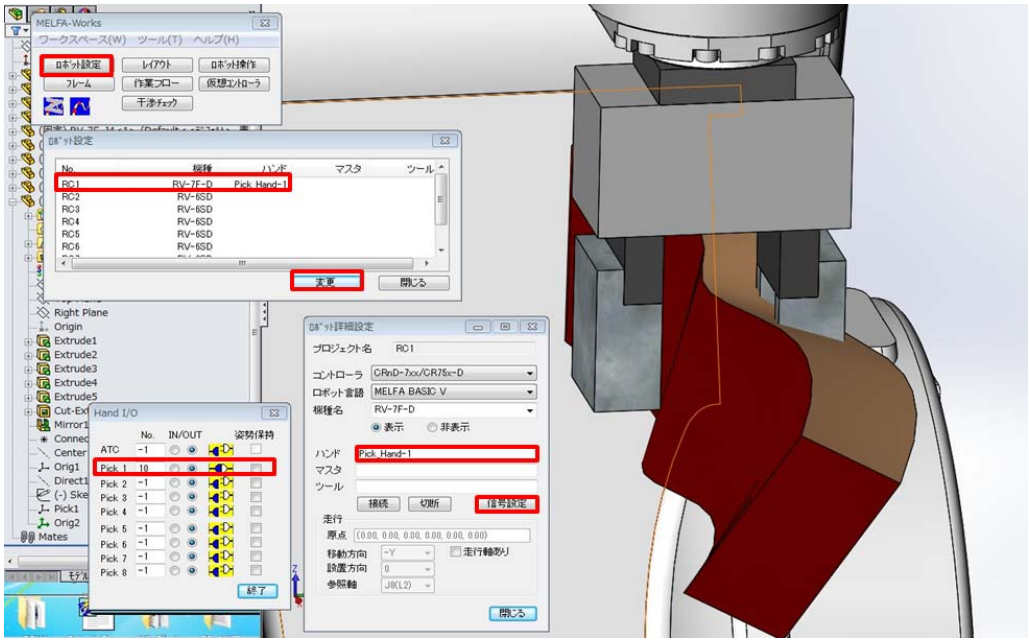


図 7-81：ワーク把持

3) 把持位置の教示

※ この作業は Ex-T スプラインの場合のみ必要となります。スプラインの場合は「4) 経路作成」の作業を実施してください。

[作業フロー] をクリックし、作業フロー画面を表示します。作業フロー画面の [取り込み] ボタンをクリックし、ワークを把持する位置を教示します。

※ ここで教示したデータが把持位置となります。

※ 把持位置を教示後にワークを動かさないようにしてください。

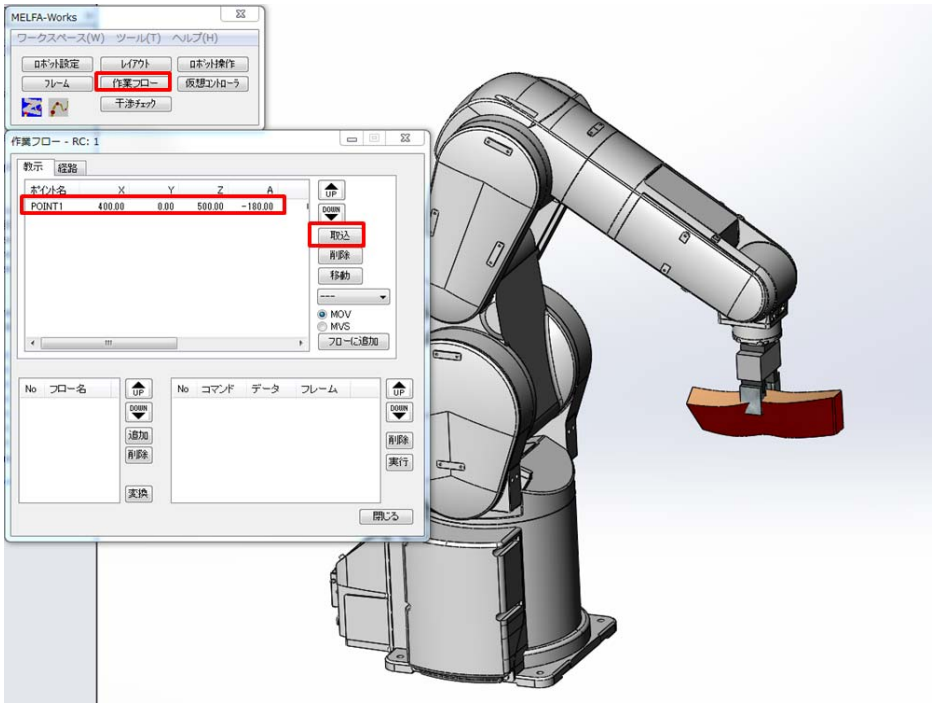


図 7-82：把持位置の教示

4) 経路作成

経路データを作成するために CAD リンク用のハンドに付け替えます。
[作業フロー] をクリックし、作業フロー画面を表示します。
作業フロー画面の経路タブを選択し、[追加] ボタンをクリックすると、加工経路が追加されます。
追加された加工経路を選択し、[編集] ボタンをクリックすると、加工設定画面が表示されます。
ワーク上の作業経路を選択し、[追加] ボタンをクリックすると、加工経路を登録することができます。
※Ex-T スプラインの場合は垂直多関節 (6 軸) ロボットでも Z 反転にチェックを入れないようにしてください。

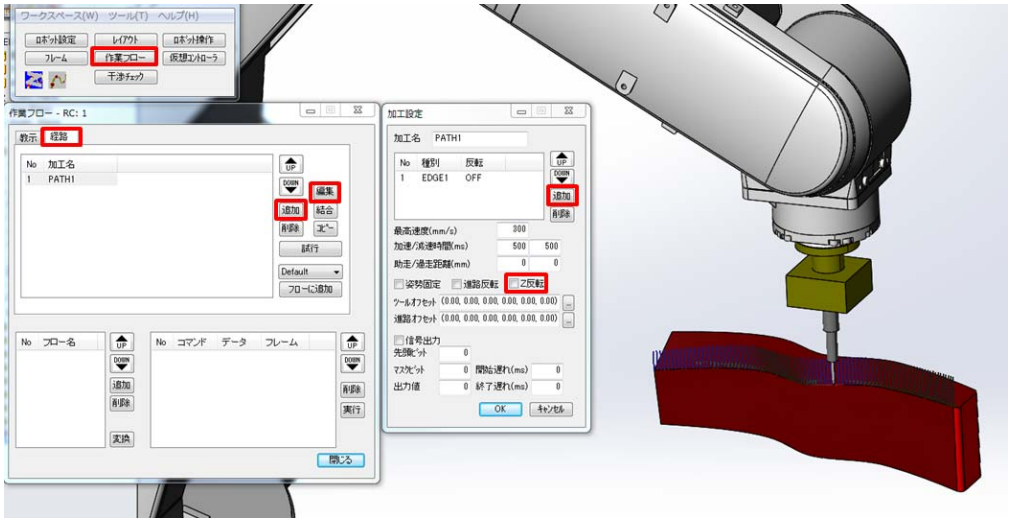


図 7-83：経路作成

5) 作業フローの作成

作業フローの作成をおこないます。
※ スプラインと Ex-T スプラインで作業フローの手順が異なるため注意してください。
[作業フロー] をクリックし、作業フロー画面を表示します。
フローの [追加] ボタンをクリックし、フローを追加します。
作業フロー画面の経路タブを選択します。
加工経路を選択し、[フローに追加] ボタンをクリックすると、選択したデータがフローに追加されます。

図 7-84：フロー作成 (加工経路登録)

Ex-T スプラインの場合は作業フローに「3) 把持位置の教示」で教示した把持位置を登録する必要があります。
作業フロー画面で教示タブを選択します。
教示した把持位置を選択し、[フローに追加] ボタンをクリックすると、選択した位置データがフローに追加されます。

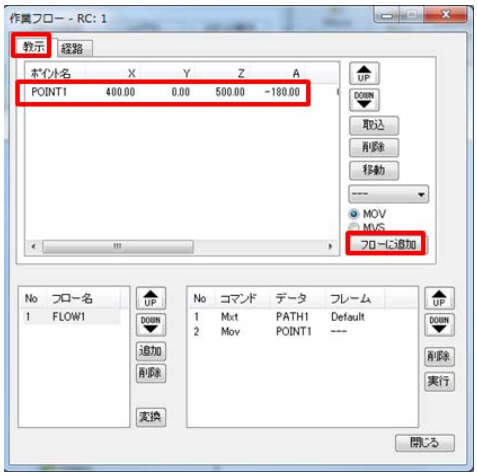


図 7-85：フロー作成 (把持位置登録)

6) MXT ファイルの出力
作成した作業フローに応じた MXT ファイルを出力します。
作業フロー画面の [変換] ボタンをクリックすると、出力先設定ダイアログが表示されます。
プログラム名に任意の名前を入力し、[OK] ボタンをクリックすると、MELFA-Works で指定したプロジェクトフォルダの下記の場所に入力したプログラム名のプログラムと MXT ファイルが作成されます。

・フォルダ構成
MXT ファイル　：MELFA-Works¥MXT01_01.MXT
プログラム　：FLOW.PRG　（※ プログラム名に ”FLOW” を設定した場合 ）

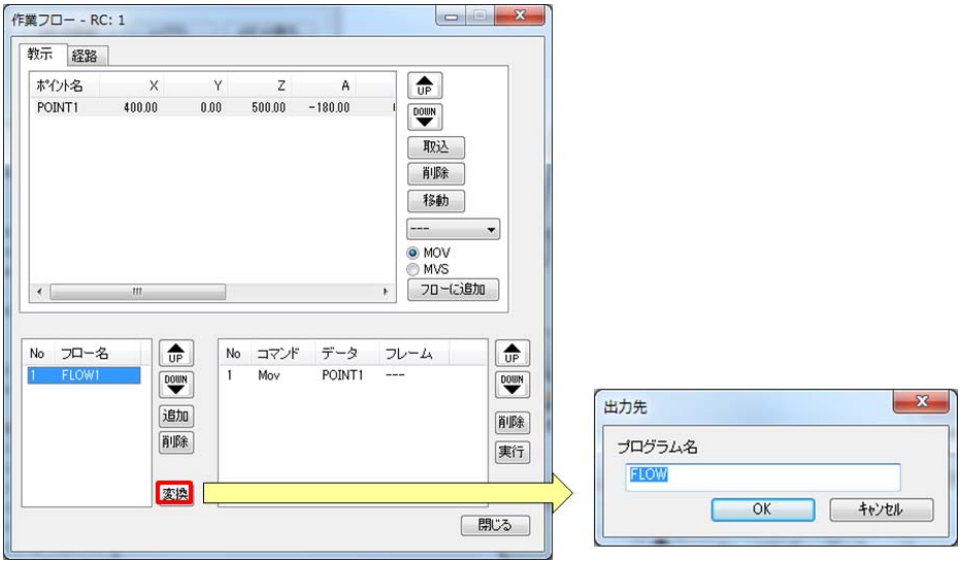


図 7-86：MXT ファイルの出力

■インポート

RT ToolBox2 Ver.3.20W 以降では、CSV 形式だけでなく MXT 形式のファイルに対応しています。

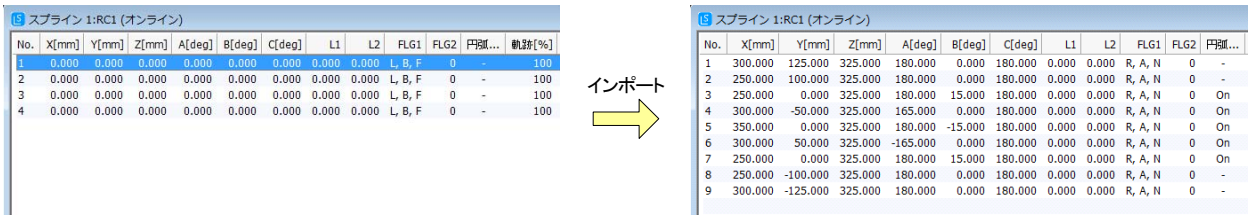


図 7-87：インポート

⚠注意　インポートにより、スプラインファイル編集画面に登録されていた全ての経路点データは、新たに取り込まれた経路点データに置き換わります。

・CSV ファイル

メニュー [ファイル] → [インポート] をクリックすると、ファイル選択ダイアログが表示されます。所望の CSV ファイルを選択して開くと、アクティブとなっているスプラインファイル編集画面に CSV ファイルの内容が取り込まれます。

※Ex-T スプラインの場合はデータインポート後に Ex-T 制御設定を行う必要があります。

・MXT ファイル

メニュー [ファイル] → [インポート] をクリックすると、ファイル選択ダイアログが表示されます。所望の MXT ファイルを選択して開くと、トレランス設定画面が表示されます。

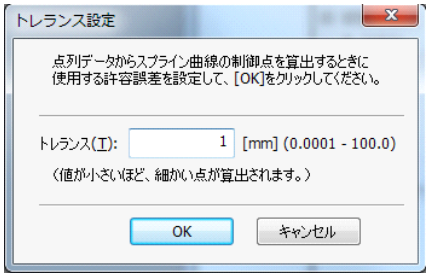


図 7-88：トレランス設定画面

点列データとスプライン曲線との許容誤差（mm）を設定して [OK] ボタンをクリックすると Ex-T 制御設定画面が表示されます。

スプラインの場合は設定を行う必要がないため、[OK] ボタンをクリックすると、アクティブとなっているスプラインファイル編集画面に MXT ファイルの内容が取り込まれます。（許容誤差が小さいほど、経路点が細かく算出されます。）

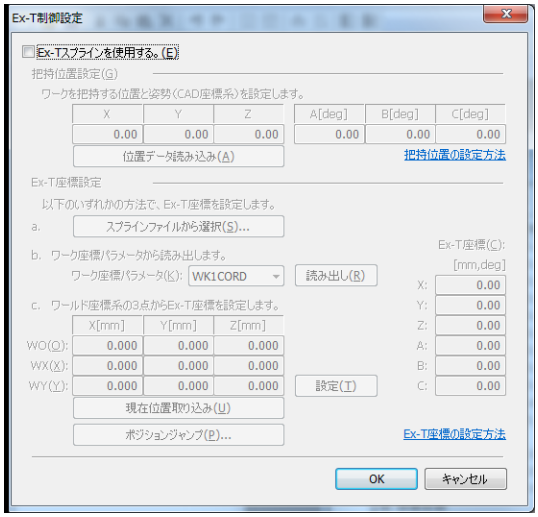


図 7-89：Ex-T 制御設定画面（初期画面）

Ex-T スプラインの場合は把持位置と Ex-T 座標の設定を行う必要があります。
[Ex-T スプラインを使用する] チェックボックスにチェックを入れます。
[位置データ読み込み] ボタンをクリックすると、プログラム選択画面が表示されますので、MXT ファイル出力時に作成したプログラムを選択します。

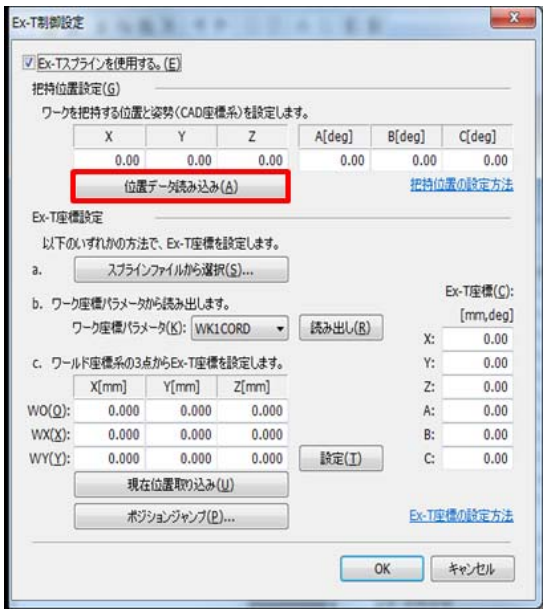


図 7-90：Ex-T 制御設定画面（設定方法）

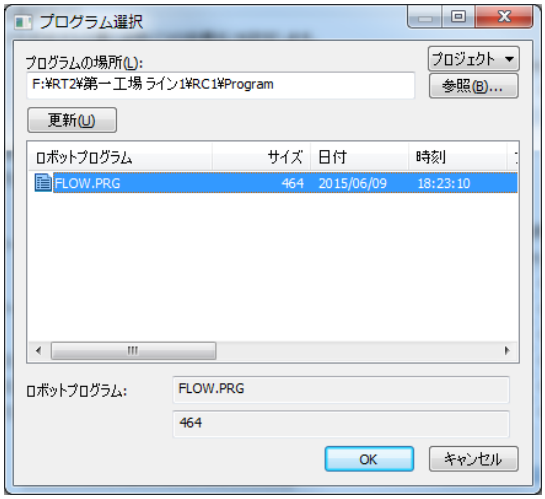


図 7-91：プログラム選択画面

プログラムを選択し、[OK] ボタンをクリックすると、直交位置データ選択画面が表示されます。

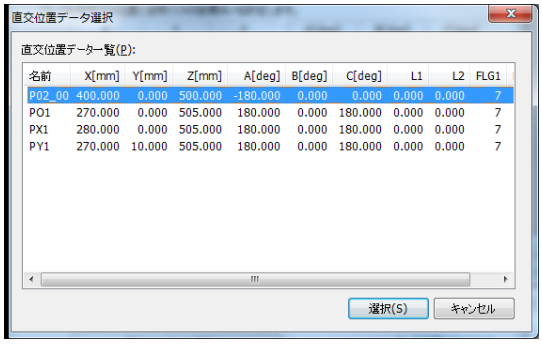


図 7-92：直交位置データ選択画面

読み込む位置データ (MXT ファイル作成時に教示した把持位置データ) を選択し、[OK] ボタンをクリックすると、把持位置が設定されます。

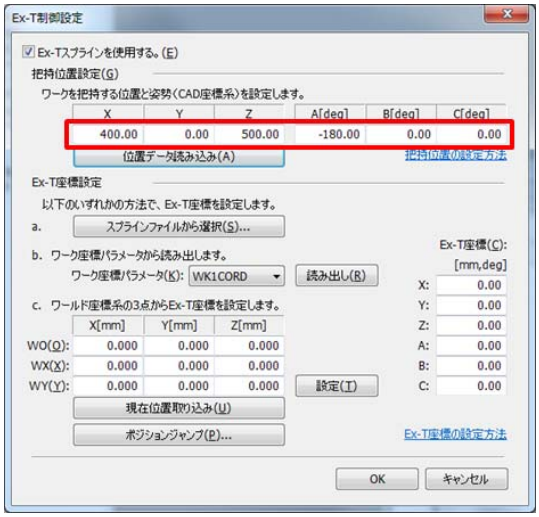


図 7-93：Ex-T 制御設定 (把持位置設定)

Ex-T 座標の設定を行い [OK] ボタンをクリックすると MXT ファイルの内容が Ex-T スプライン用の経路点データに変換され取り込まれます。

⚠注意　対応する点列のデータ型は、直交データのみです。
MXT ファイルのヘッダ情報の移動速度（加減速時間・最高速度）はインポートされないため、ロボットプログラムの MvSpl 命令の引数に反映する必要があります。

⚠注意　トレランスの値を小さくすると算出される経路点の精度は上がりますが、経路点間の距離が短くなるため速度が出せなくなります。

■エクスポート
　メニュー[ファイル]→[エクスポート] をクリックすると、CSV ファイルの保存先を指定するダイアログが表示されます。保存先のフォルダと CSV ファイルの名前を指定して保存すると、アクティブとなっているスプラインファイル編集画面の経路点データの内容が CSV ファイルに書き出されます。

(15) 編集の補助機能
■カット＆ペースト
　指定された経路点データを切り取り、別の経路点データへ貼り付けます。
　経路点データー覧エリアで切り取り元となる経路点データを選択し、メニュー[編集]→[切り取り] をクリックします。次に貼り付け先となる経路点データを選択し、メニュー[編集]→[貼り付け] をクリックすると、[切り取り] 操作で記憶した経路点データが貼り付け先の経路点データへ順に貼り付けられます。それと同時に切り取り元の経路点データは一覧から削除されます。

図 7-94：カット＆ペースト

◇◆◇貼り付け先の経路点データの数が少ない場合◇◆◇
　切り取り元として指定した経路点データの数に対し、貼り付け先以降の経路点データの数が少ない場合、貼り付け先のない経路点データは貼り付けられません。

■コピー＆ペースト
　指定された経路点データの内容を別の経路点データへコピーします。
　経路点データー覧エリアでコピー元となる経路点データを選択し、メニュー[編集]→[コピー] をクリックします。次にコピー先となる経路点データを選択し、メニュー[編集]→[貼り付け] をクリックすると、[コピー] 操作で記憶した経路点データがコピー先の経路点データへ順にコピーされます。

図 7-95：コピー＆ペースト

◇◆◇コピー先の経路点データの数が少ない場合◇◆◇
　　コピー元として指定した経路点データの数に対し、コピー先以降の経路点データの数が少ない場合、コピー先のない経路点データはコピーされません。

■コピー＆ペースト（位置データ）
　指定された位置データの内容をスプラインファイルの経路点データへコピーします。
　ロボットプログラムでコピー元となる位置データを選択し、メニュー［編集］→［コピー－位置データ］をクリックします。次にコピー先となる経路点データを選択し、メニュー［編集］→［貼り付け－位置データ］をクリックすると、［コピー－位置データ］操作で記憶した位置データがコピー先の経路点データへ順にコピーされます。

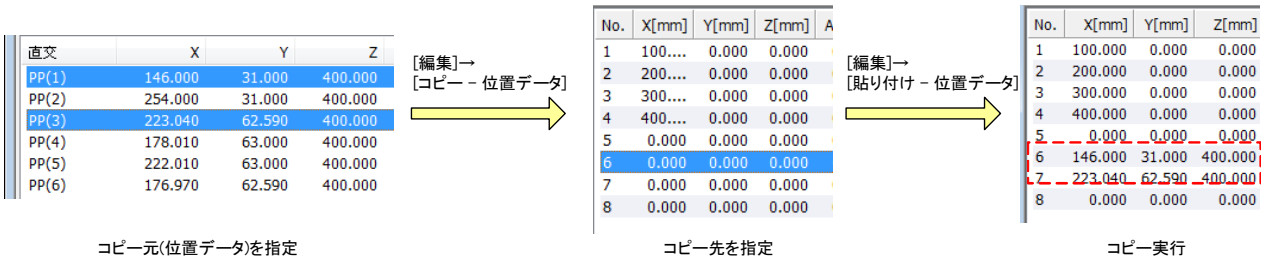


図 7-96：コピー＆ペースト（位置データ）

■コピーした経路点の挿入
　指定された経路点データの内容を別の行へ新たな経路点データとして挿入します。
　経路点データー覧エリアでコピー元となる経路点データを選択し、メニュー［編集］→［コピー］をクリックします。次に挿入先となる経路点データを選択し、メニュー［編集］→［コピーした経路点の挿入］をクリックすると、挿入先として選択した経路点データの直前に［コピー］操作で記憶した経路点データが順に挿入されます。

図 7-97：コピーした経路点の挿入

■元に戻す（アンドゥ）
　メニュー［編集］→［元に戻す］をクリックすると、経路点データに対して最後に適用された編集内容を取り消して、適用前の状態に戻します。具体的には、以下の操作が対象となります。
　　・経路点データの追加・挿入・削除
　　・経路点データの変更（編集内容を適用）
　　・経路点データのカット＆ペースト・コピー＆ペースト・コピーした経路点の挿入
　　・インポート
　　・位置補正機能
　　・フレーム変換（変換実行）

■やり直し（リドゥ）
　メニュー［編集］→［やり直し］をクリックすると、［元に戻す］で取り消した編集内容を再び経路点データへ適用します。

(16) スプライン曲線の表示

スプライン曲線を表示します。この機能は、RT ToolBox2 Ver.3.20W 以降で使用できます。

メニュー [ツール] → [スプライン曲線] をクリックすると、3D モニタ画面が開いて編集中のスプラインファイルのスプライン曲線が 3D モニタ画面に表示されます。

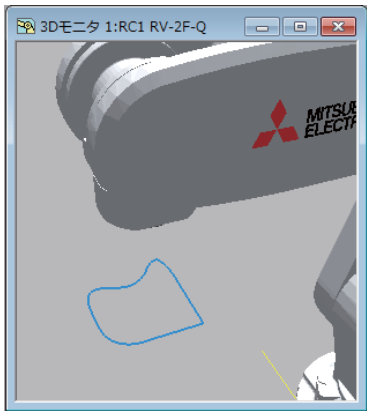


図 7-98：スプライン曲線の表示

3D モニタ画面の使用方法については、「RT ToolBox2/RT ToolBox2 mini 取扱説明書」を参照願います。

(17) 編集内容の確認

■経路点チェック

事前に各経路点に対して、経路点間の距離・姿勢の変化量・構造フラグの値などをチェックします。

この機能は、RT ToolBox2 Ver.3.20W 以降で使用できます。

メニュー [ツール] → [経路点チェック] をクリックすると、アクティブになっているスプライン編集画面の経路点チェック画面が表示されます。

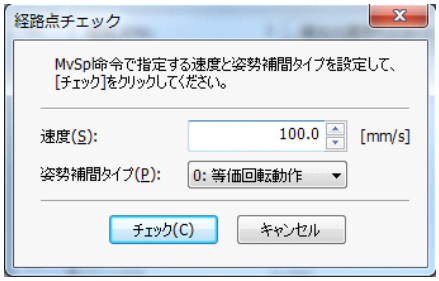


図 7-99：経路点チェック画面

MvSpl 命令 /EMvSpl 命令で指定する速度と姿勢補間タイプを設定して [チェック] ボタンを押すと、各経路点のチェックが実行されます。正常に終了すると完了ダイアログが表示されます。エラーが発生した場合は、エラーダイアログが表示されます。エラー内容については、別冊の「トラブルシューティング」を参照ください。

※ 経路点チェックでは、オーバーライド 100% で動作させた場合の速度でチェックされます。

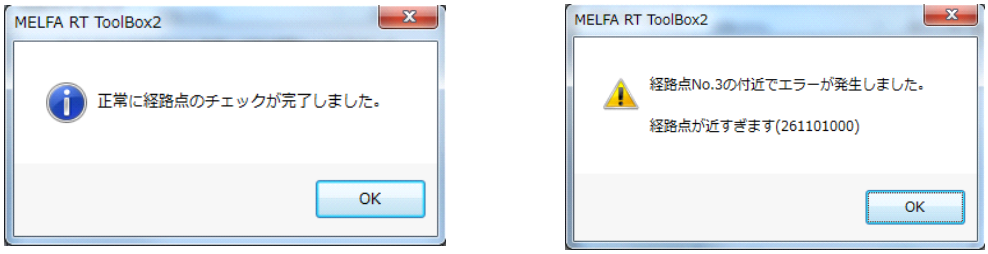


図 7-100：経路点チェック完了ダイアログ（左）/ エラーダイアログ（右）

■動作範囲チェック
　事前に各経路点に対して、ロボットが動作可能かをチェックします。
　この機能は、RT ToolBox2 Ver.3.40S 以降のシミュレーション機能のみ使用できます。
　メニュー[ツール]→[動作範囲チェック]をクリックすると、アクティブになっているスプライン編集画面の各経路点への動作が実行されます。正常に終了すると完了ダイアログが表示されます。エラーが発生した場合は、エラーダイアログが表示されます。エラー内容については、別冊の「トラブルシューティング」を参照ください。
　※ 動作範囲チェックでは、ロボットの軌跡は保証されません。

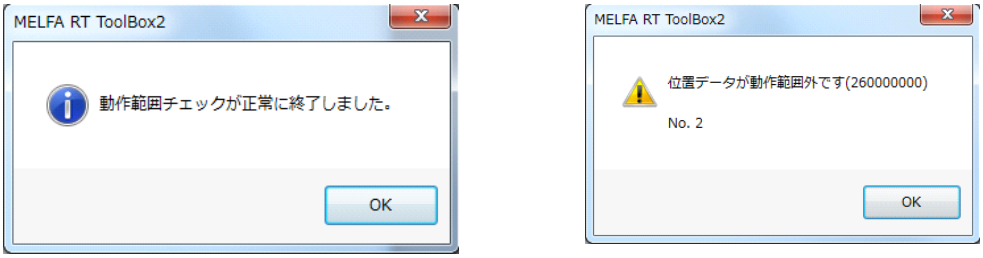


図 7-101：動作範囲チェック完了ダイアログ（左）/ エラーダイアログ（右）

■最大速度算出
　事前に経路点間の距離からスプライン補間命令（MvSpl 命令、EM v Spl 命令 ）実行時にエラーとならずに実行が可能となる最大速度を算出します。
　この機能は、RT ToolBox2 Ver.3.40S 以降で使用できます。
　メニュー[ツール]→[最大速度算出]をクリックすると、アクティブになっているスプライン編集画面の最大速度算出結果が表示されます。

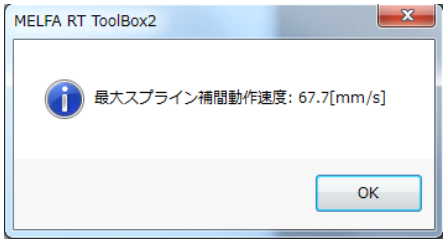


図 7-102：最大速度算出結果

(18) ロボットプログラム
■プログラム変換
　スプラインファイルに登録されている経路点データを使用したロボットプログラムを作成します。
　この機能は、RT ToolBox2 Ver.3.40S 以降で使用できます。
　メニュー[ツール]→[ロボットプログラム]→[プログラム変換]をクリックすると、アクティブになっているスプライン編集画面の新しいロボットプログラムダイアログが表示されます。

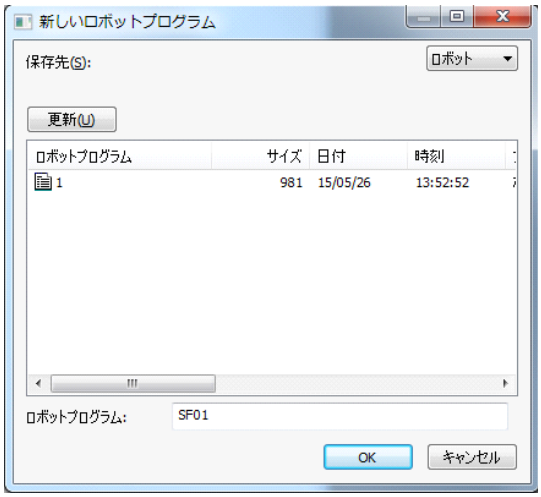


図 7-103：新しいロボットプログラム画面

保存するロボットプログラム名を入力して [OK] ボタンをクリックすると、速度設定ダイアログが表示されます。

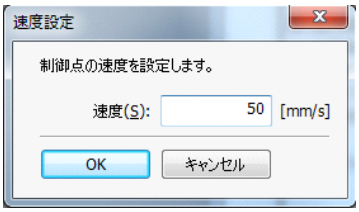
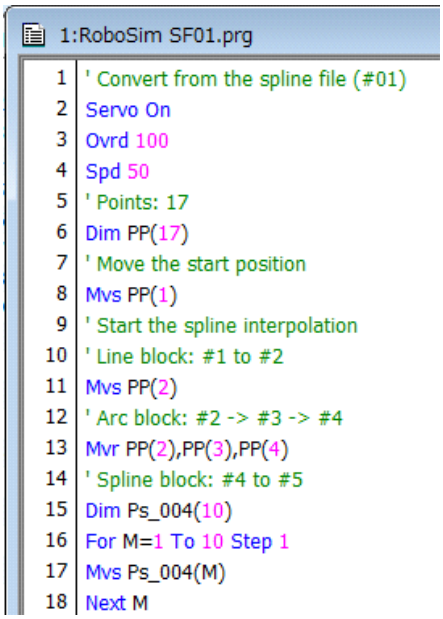


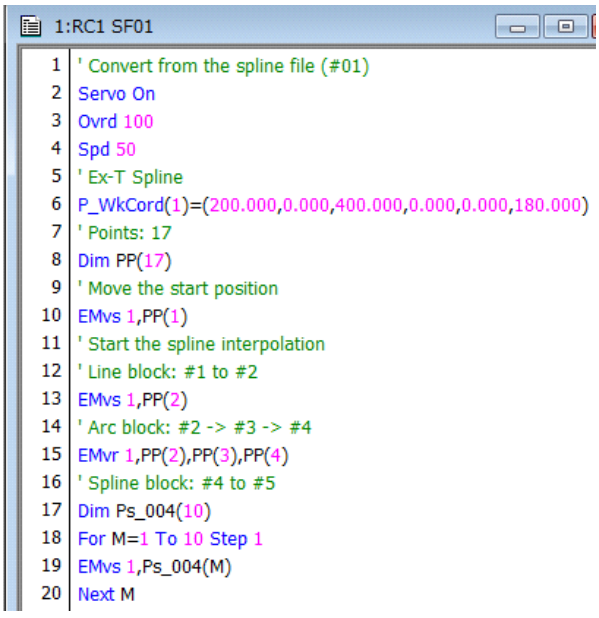
図 7-104：速度設定ダイアログ

制御点の速度を設定して [OK] ボタンをクリックすると、スプラインファイルの内容がロボットプログラムに変換されます。



```
1:RoboSim SF01.prg
1  ' Convert from the spline file (#01)
2  Servo On
3  Ovrd 100
4  Spd 50
5  ' Points: 17
6  Dim PP(17)
7  ' Move the start position
8  Mvs PP(1)
9  ' Start the spline interpolation
10 ' Line block: #1 to #2
11 Mvs PP(2)
12 ' Arc block: #2 -> #3 -> #4
13 Mvr PP(2),PP(3),PP(4)
14 ' Spline block: #4 to #5
15 Dim Ps_004(10)
16 For M=1 To 10 Step 1
17 Mvs Ps_004(M)
18 Next M
```

スプライン補間



```
1:RC1 SF01
1  ' Convert from the spline file (#01)
2  Servo On
3  Ovrd 100
4  Spd 50
5  ' Ex-T Spline
6  P_WkCord(1)=(200.000,0.000,400.000,0.000,0.000,180.000)
7  ' Points: 17
8  Dim PP(17)
9  ' Move the start position
10 EMvs 1,PP(1)
11 ' Start the spline interpolation
12 ' Line block: #1 to #2
13 EMvs 1,PP(2)
14 ' Arc block: #2 -> #3 -> #4
15 EMvr 1,PP(2),PP(3),PP(4)
16 ' Spline block: #4 to #5
17 Dim Ps_004(10)
18 For M=1 To 10 Step 1
19 EMvs 1,Ps_004(M)
20 Next M
```

Ex-T スプライン補間

図 7-105：各補間から作成されるプログラム

■位置データ取り込み

ロボットプログラムの位置データをスプラインファイルの経路点データとして取り込むことができます。

この機能は、RT ToolBox2 Ver.3.40S 以降で使用できます。

メニュー [ツール] → [ロボットプログラム] → [位置データ取り込み] をクリックすると、アクティブになっているスプライン編集画面に取り込むプログラムの選択画面が表示されます。

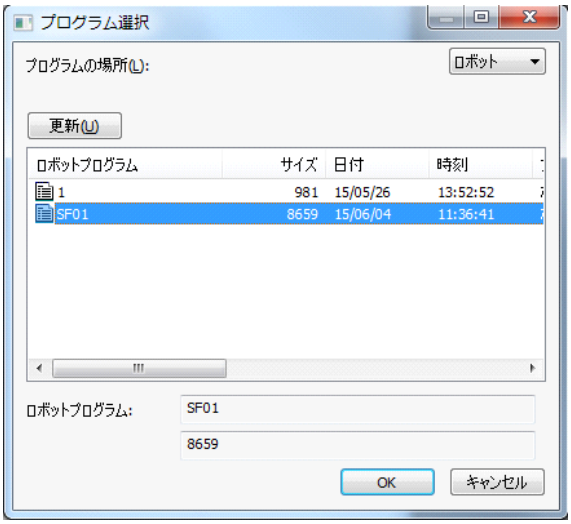


図 7-106：プログラム選択画面

位置データを取り込むロボットプログラムを選択して [OK] ボタンをクリックすると、アクティブになっているスプライン編集画面に位置データが取り込まれます。

※ 取り込むことのできる位置データは PP(n) (n：経路点番号) または P* (*：経路点番号) となっています。

※ アクティブになっているスプラインファイルに対応する経路点が存在しない場合は、経路点を新規に作成します。(位置データ以外は、初期値となります。)

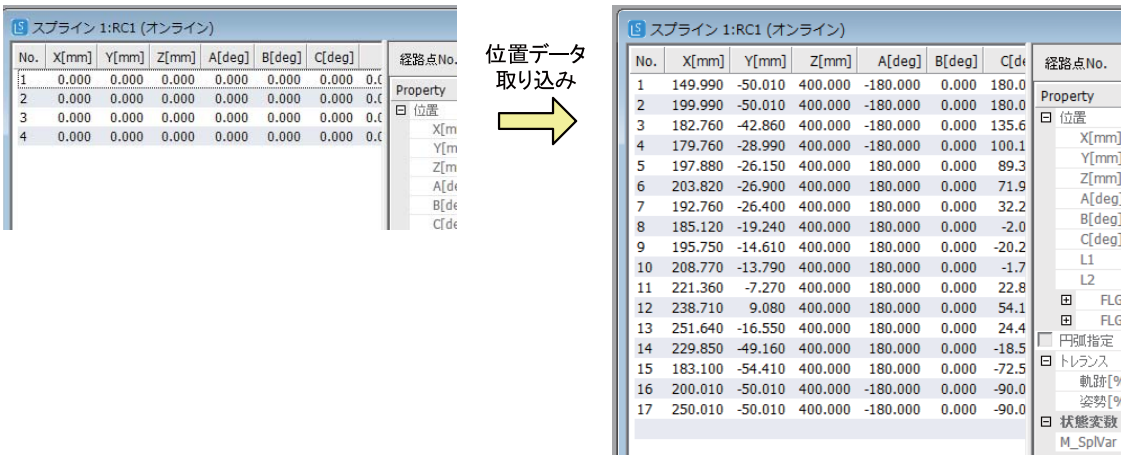


図 7-107：位置データ取り込み

7.4.6 ロボットプログラムの作成

RT ToolBox2・ティーチングボックスのプログラム編集画面でロボットプログラムを作成します。

スプライン補間を実行するには、表 7-38 と表 7-40 に記載されている命令・ロボット状態変数を使用します。詳細につきましては、164 ページの「4.13 命令の詳細説明」・288 ページの「4.14 ロボット（システム）状態変数の詳細解説」を参照してください。

表 7-38：スプライン補間で使用する命令

命令語	解説	参照ページ
MvSpl（ムーブスプライン）	スプライン補間を実行します。	245
EMvSpl（イームーブスプライン）	Ex-T スプライン補間を実行します。	212
SetCalFrm（セット キャリブレーションフレーム）	フレーム変換で使用する座標系を設定します。	267

表 7-39：スプライン補間で使用する組込み関数

関数名称	解説	参照ページ
SplPos	指定したスプラインファイルに登録されている経路点データを位置変数に代入します。	401
SplSpd	指定したスプラインファイルの経路点データに基づき、MvSpl 命令 /EMvSpl 命令実行時に L2611（経路点が近すぎます）エラーが発生しない速度を数値変数に代入します。	402
SplECord	指定したスプラインファイルに登録されている Ex-T 座標系原点データを位置変数に代入します。	400

表 7-40：スプライン補間に関係するロボット状態変数

変数名称	配列要素数	内容	属性[注 1)]	データの型	参照ページ
M_SplPno	メカ番号(1 ～ 3)	最も新しい通過済みの経路点の番号を返します。	R	整数型	335
M_SplVar	メカ番号(1 ～ 3)	最新の経路点データの数値設定値を返します。 書き込みにより、値を変更することもできます。	RW	整数型	336

注 1）R.....読み出しのみ可能です。
　　RW.....読み出し、書き込みとも可能です。

■サンプルプログラム

経路 1 の経路点データがスプラインファイル 05 に登録されています。フレーム変換を使用して、経路 1 と経路 2 のスプライン補間を実行します。その際、スロット 2 のプログラムで経路点データの数値設定を利用して、動作途中に出力信号 100 番と 101 番の信号を ON/OFF します。

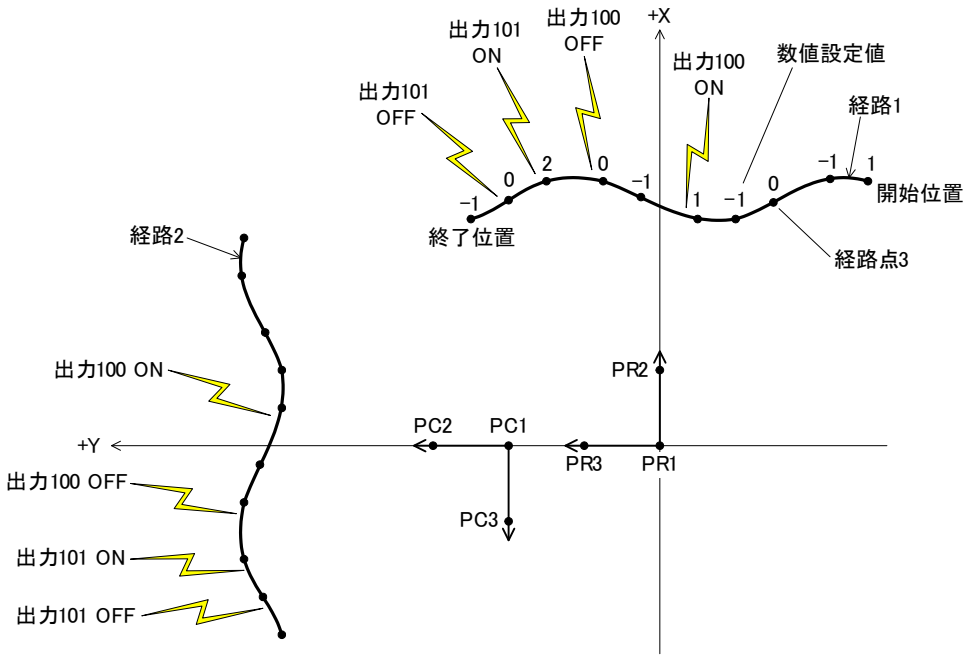


図 7-108：サンプルプログラムの動作

スロット 1

```
Ovrd 100
Mov P1                                  ‘ 待機位置（P1）へ関節補間で移動
 ‘
Fine 200                                ‘ 位置決めパルスを 200 に変更
Spd 100                                 ‘ 開始位置までの直線補間速度を 100mm/s に指定
M_00=1                                  ‘ スロット 2 同期用のフラグを ON
MvSpl 5, 50, 10                         ‘ 経路 1 を加減速距離 10mm、速度 50mm/s でスプライン補間
M_00=0                                  ‘ 同期用フラグを OFF
Fine 0                                  ‘ 位置決めパルス指定を無効
Mvs P1                                  ‘ 待機位置へ直線補間で移動
 ‘
PR1=(0, 0, 0, 0, 0, 0)(0, 0)            ‘ フレーム変換用の基準座標系の位置を設定（PR1 ～ PR3）
PR2=(20, 0, 0, 0, 0, 0)(0, 0)
PR3=(0, 20, 0, 0, 0, 0)(0, 0)
PC1=(0, 40, 0, 0, 0, 0)(0, 0)           ‘ 変換後の基準座標系の位置を設定（PC1 ～ PC3）
PC2=(0, 60, 0, 0, 0, 0)(0, 0)           ‘ （変換前に対して Z 軸周りに 90 度回転・Y 軸方向に 40mm
PC3=(-20, 40, 0, 0, 0, 0)(0, 0)         ‘ 移動した座標系）
SetCalFrm PR1, PR2, PR3, PC1, PC2, PC3  ‘ フレーム変換用の座標系を算出し設定
 ‘
Fine 200
M_00=1
MvSpl 5, 50, 10, 2                      ‘ スプライン番号 5 の経路点に対してフレーム変換を実施し、
                                        ‘ 経路 2 をスプライン補間
M_00=0
Fine 0
Mvs P1
End
```

スロット 2

```
Def IO PORT1=Byte, 100, &H03            ' 変数 PORT1 に出力信号 100 と 101 を割り当てる
M_SplVar=0                              ' M_SplVar の値を 0 リセット
Wait M_OO=1                             ' スプライン補間が開始するまで待つ
*L1:If M_SplPno<3 Then GoTo *L1         ' 経路点 3 を通過するまで待つ
*L2
Select M_SplVar
Case 1                                  ' 数値設定値に 1 が設定された経路点を通過
PORT1=1                                 ' 出力信号 100 を ON
Break
Case 2                                  ' 数値設定値に 2 が設定された経路点を通過
PORT1=2                                 ' 出力信号 101 を ON
Break
Default                                 ' 数値設定値が 1、2 ではない
PORT1=0                                 ' 出力信号 100 と 101 を OFF
Break
End Select
If M_OO=1 Then Goto *L2                 ' スプライン補間が終了するまで繰り返す
End
```

7.4.7 動作の確認

RT ToolBox2 のシミュレーション機能を使用して、作成したロボットプログラムのスプライン補間の動作を確認します。所望の動作と異なる場合は、経路点データの設定・ロボットプログラムを見直し修正します。

シミュレーション機能の使用方法については、「RT ToolBox2/RT ToolBox2 mini 取扱説明書」を参照願います。

なお、RT ToolBox2 mini ではシミュレーション機能は使用できません。

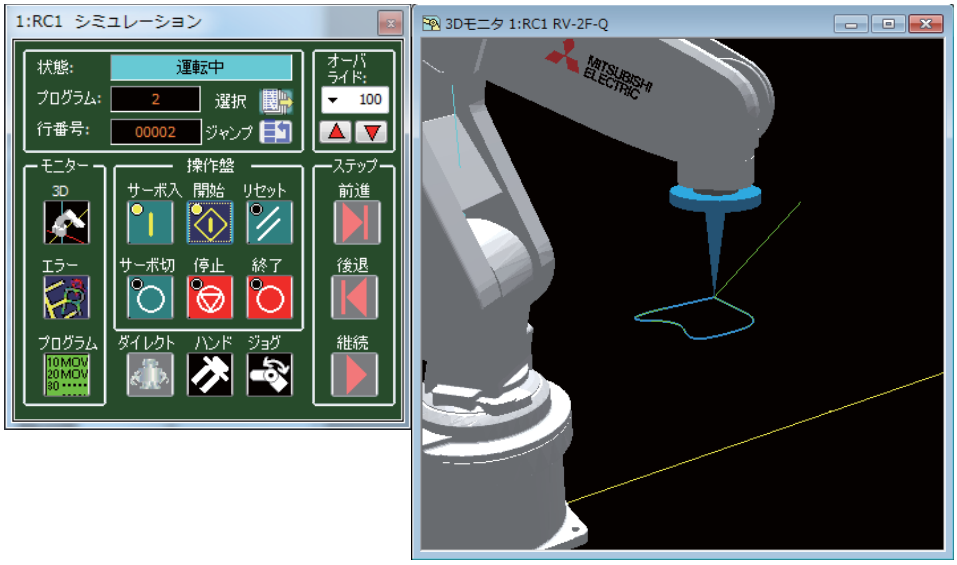


図 7-109：シミュレーション機能

◇◆◇シミュレーションで使用するロボットプログラムとスプラインファイル◇◆◇

シミュレーションでスプライン補間を実行するには、ロボットプログラムとスプラインファイルが RT ToolBox2 内の仮想コントローラに登録されている必要があります。シミュレーションが起動しているプロジェクトの［オンライン］の中にロボットプログラムとスプラインファイルが登録されていない場合は、以下の方法で仮想コントローラへコピーしてください。

・ロボットプログラム：RT ToolBox2 のプログラム管理を使用します。

・スプラインファイル：対象ファイルをスプラインファイル編集画面で開き、622 ページの「(9) スプラインファイルの保存」を行います。または、625 ページの「(12) スプラインファイルのコピー」を行います。

7.4.8 ロボットコントローラへ保存

ロボットプログラムとスプラインファイルをロボットコントローラへ保存します。

スプラインファイルの保存方法については、622 ページの「(9) スプラインファイルの保存」を参照してください。

7.4.9 調整作業

実際のシステムにおいて、デバッグ操作（ステップ送り）によりスプライン補間の動作を確認します。

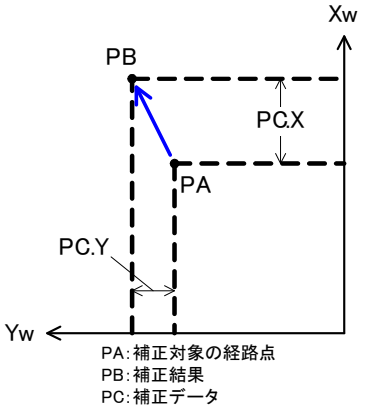
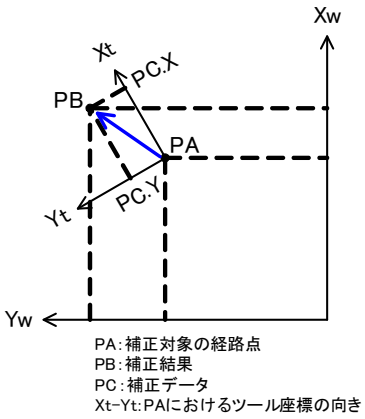
所望の動作と異なる場合は、経路点データの設定・ロボットプログラムを見直し修正します。経路点データを修正するには、スプラインファイルを RT ToolBox2 のスプラインファイル編集画面へ読み出し、経路点データの設定値を変更した後、コントローラへ書き込みます。

ここでは、位置データを補正するために RT ToolBox2 に用意されている「位置補正機能」・「フレーム変換機能」・「ポジションジャンプ機能」と、パラメータ SPLOPTGC（アクティブゲイン制御のゲイン補正率）について説明します。

(1) 位置補正機能

経路点データのロボットの位置に対して、MELFA-BASIC V の位置データの相対演算と同様の補正を実施することができます。表 7-41 に示すように補正方法には 2 種類の方法があります。

表 7-41：位置の補正方法

補正方法	説明
ベクトル和演算（P+P）	経路点データのロボットの位置データに対して、補正データの値を加算（各座標成分同士の足し算）します。ワールド座標系に沿った補正となります。 構造フラグ・多回転フラグ・付加軸データは元の値から変更されません。  ベクトル和演算(P+P)
ベクトル積演算（PxP）	経路点データのロボットの位置データに対して、補正データの値を乗算します。 （補正後の位置 = 経路点の位置 × 補正データ） ツール座標系に沿った補正となります。 構造フラグ・多回転フラグ・付加軸データは元の値から変更されません。  ベクトル積演算(PxP)

位置補正機能を使用するには、メニュー [ツール] → [位置補正] をクリックします。[位置補正] をクリックすると、位置補正画面が表示されます。

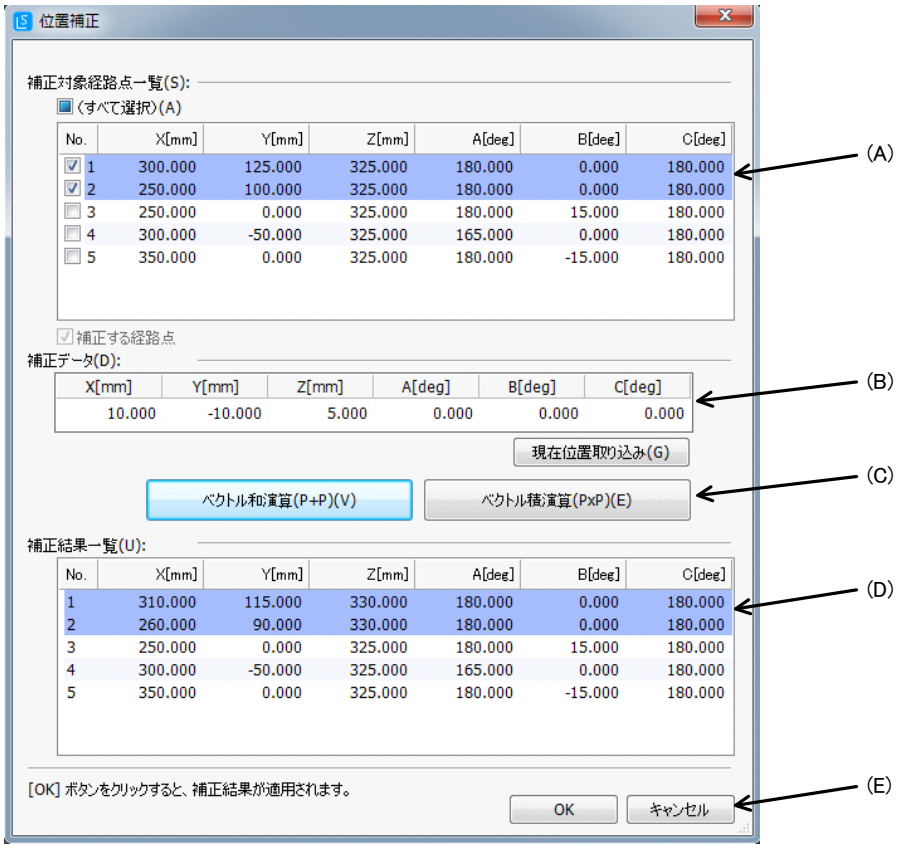


図 7-110：位置補正画面

位置補正は次の手順でおこないます。

(A)「補正対象経路点一覧」にアクティブとなっているスプラインファイル編集画面の経路点データが表示されます。この中から、位置補正を実施する経路点データにチェックを入れて選択します。複数の経路点データを選択することができます。

(B) 補正データを設定します。
コントローラと接続されている場合は [現在位置取り込み] のボタンが表示され、このボタンを押すとロボットの現在位置が補正データとして取り込まれます。

(C) 補正方法を選択し、ボタンを押します。

(D)「補正結果一覧」に補正実施後の経路点データが表示されます。補正の対象として選択されていなかった経路点データは元の値と同じです。

(E)[OK] ボタンを押すと、補正の結果をスプラインファイル編集画面の経路点データに反映して位置補正画面を閉じます。
[キャンセル] ボタンを押すと、補正の結果を破棄して位置補正画面を閉じます。

(2) フレーム変換機能

メニュー [ツール] →[フレーム変換] をクリックすると、アクティブとなっているスプラインファイル編集画面の経路点データに対してフレーム変換を実施するための画面が表示されます。

（フレーム変換については 598 ページの「(5) フレーム変換」を参照してください。）

図 7-111：フレーム変換画面

■座標系の設定

図 7-111 (A) の部分で、基準座標系と変換後の基準座標系を定義するそれぞれ 3 つの位置（原点・X 軸上の位置・X-Y 平面の +Y 方向の位置）の X,Y,Z 軸座標値を設定します。

基準座標系については、[経路点一覧から選択] ボタンを押すと、登録されている経路点データの中から設定する座標値を選択することができます。

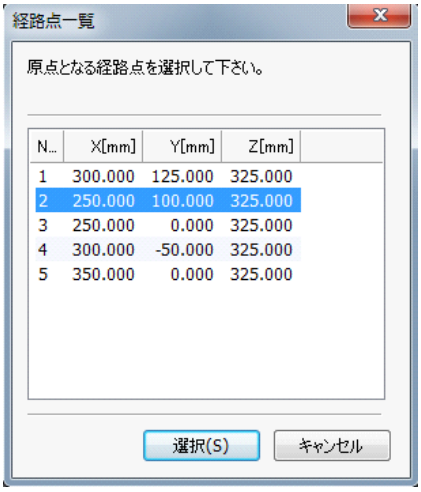


図 7-112：経路点一覧からの選択

コントローラと接続されている場合は、[現在位置取り込み] ボタンが表示されます。このボタンを押すと、選択している変換後の基準座標系の位置にロボットの現在位置が設定されます。

■ RT ToolBox2 でフレーム変換を実行

座標系を設定した後、図 7-111 (B) の [変換実行] ボタンを押すと、全ての経路点データに対してフレーム変換を実施し、その結果を図 7-111(C) の「変換結果一覧」画面に表示します。

フレーム変換を実施すると、図 7-111(D) の [変換結果を適用して終了] ボタンが有効となります。

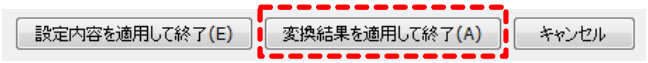


図 7-113：[変換結果を適用して終了] ボタン

このボタンを押すと、変換結果をスプラインファイル編集画面の経路点データに反映してフレーム変換画面を閉じます。
[キャンセル] ボタンを押すと、変換結果を破棄してフレーム変換画面を閉じます。
なお、座標系の設定が適切ではない（同一点・3 点が直線上）場合は、「フレーム設定が正しくありません。」とダイアログが表示され、フレーム変換を実施することはできません。

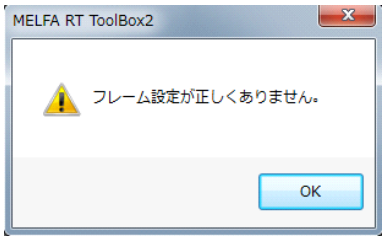


図 7-114：座標系の設定が正しくない場合のダイアログ（変換実行時）

■座標系をスプラインファイルに保存
座標系を設定した後、図 7-111 (D) の [設定内容を適用して終了] ボタンを押すと、座標系を定義する位置データをスプラインファイル編集画面に記憶してフレーム変換画面を閉じます。
MvSpl 命令 /EMvSpl 命令の引数 < フレーム変換 > に ”1” を指定した場合、スプライン補間実行時にここで設定した座標系を使用してフレーム変換をおこないます。
[キャンセル] ボタンを押すと、座標系の設定を破棄してフレーム変換画面を閉じます。（既にスプラインファイル編集画面に座標系が記憶されている場合、その内容は変更されません。）
なお、座標系の設定が適切ではない（同一点・3 点が直線上）場合は、座標系の設定が正しくないことを示すダイアログが表示されます。

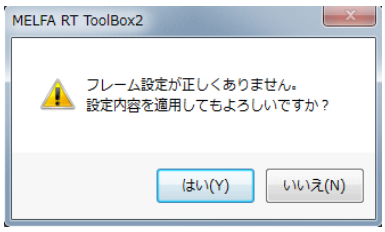


図 7-115：座標系の設定が正しくない場合のダイアログ（座標系保存時）

[はい] ボタンを押すと、その設定をスプラインファイル編集画面に記憶してフレーム変換画面を閉じます。
[いいえ] ボタンを押すと、スプラインファイル編集画面には記憶せずフレーム変換画面に戻ります。

■座標系の設定クリア
スプラインファイル編集画面に記憶された座標系の設定を未設定状態とするには、図 7-111 (E) の [クリア] ボタンを押します。このボタンを押すと確認ダイアログが表示されます。

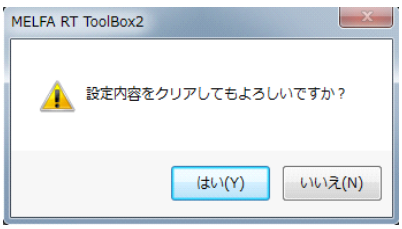


図 7-116：座標系クリアの確認ダイアログ

［はい］ボタンを押すと、スプラインファイル編集画面に記憶された座標系の設定が未設定状態になります。
［いいえ］ボタンを押すと、設定を変更せずにフレーム変換画面に戻ります。

(3) ポジションジャンプ
選択されている経路点にポジションジャンプします。
この機能は、RT ToolBox2 Ver.3.20W 以降で使用できます。
プログラムをデバッグ状態で開いてから、経路点データ一覧エリア下部の［ポジションジャンプ］ボタンを押すと、ポジションジャンプ画面が表示されます。（メニュー［ツール］→［ポジションジャンプ］をクリックでも表示できます。）

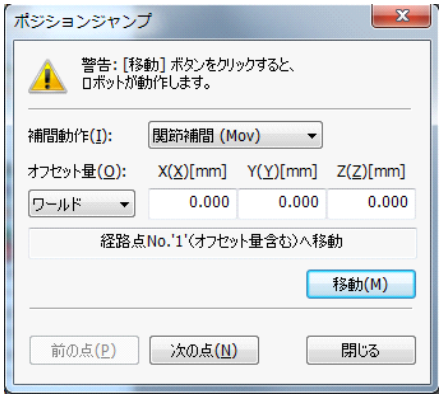


図 7-117：ポジションジャンプ画面

［次の点］（［前の点］）ボタンを押すと、対象の経路点が次の点（前の点）に切り替わります。
［移動］ボタンを押すと、選択された経路点の位置（オフセット量を含む）に指定された補間動作でロボットが移動します。
※RT ToolBox2 Ver.3.40S 以降ではオフセット量は、ワールド座標系に沿った補正量、ツール方向に沿った補正量から選択できます。

⚠注意　実際のシステムと同じ速度でロボットが動作しますので、ロボット周辺の安全にご注意ください。
いつでも非常停止できる状態でご使用ください。

対象のプロジェクトがオフラインモードの場合、［ポジションジャンプ］ボタンは表示されません。（同様に、メニュー［ツール］→［ポジションジャンプ］をクリックできません。）

(4) パラメータ SPLOPTGC

パラメータ SPLOPTGC により、スプライン補間で動作するときのロボットの制御特性を調整することができます。

通常は出荷時設定値から値を変更する必要はありません。

曲線部分の動作軌跡について、指令に対する内回り量をさらに改善したい場合は設定値を大きくしてください。設定値を大きくすると、軌跡精度が改善される可能性があります。

ロボットの揺れや振動を抑えたい場合は、逆に設定値を小さくすると改善される可能性があります。

表 7-42：パラメータ SPLOPTGC

パラメータ	パラメータ名	配列数 文字数	内容説明	出荷時 設定値
アクティブゲイン制御のゲイン補正率	SPLOPTGC	整数 1	スプライン補間実行時のロボットの制御特性（サーボの応答性）を調整することができます。設定値を大きくすると軌跡精度を改善、設定値を小さくすると振動を抑制する可能性があります。設定範囲は 1 ～ 200[%] です。 ※ 大きな値を設定すると、モータの発振やロボットが振動する場合があります。逆に小さな値を設定すると、ロボットが動作できずにサーボエラーが発生する場合があります。少しずつ値を変更して挙動を確認し、異常が起きない範囲で設定してください。 ※ 力覚制御が有効となっている場合は機能しません。 ※ 本パラメータは設定値の変更後、電源を再投入する必要がありません。	100

7.5 Ex-T制御について

7.5.1 概要

(1) 特徴

Ex-T 制御（イクスティ制御）は、外部に固定した座標系の原点をロボットの制御点として、ロボットを動作させる機能です。以下のような用途で使用します。

・研磨作業
加工対象となるワークをロボットが把持し、固定設置されたグラインダや研磨ベルトにワークを押し当てることでバリ取りや表面仕上げをおこないます。
・塗布作業
ロボットがワークを把持して、固定設置されたディスペンサから溶剤や接着剤をワークに塗布する作業。

図 7-118：研磨作業の例

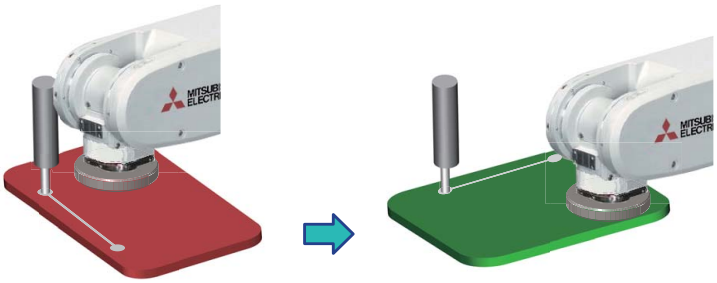


図 7-119：塗布作業の例

上図のように固定されたツール（グラインダ / ディスペンサなど）に対して、ロボットが把持したワークを動作させることによって加工（バリ取り / 研磨 / シーリングなど）をおこなう場合、ツールとワークの相対位置を保ちながら、指定した加工経路に沿った動作プログラムを生成するのは非常に困難です。

Ex-T 制御は、これらの作業やプログラミングを容易に実現するための機能です。グラインダやディスペンサの位置をロボットに登録し、その位置を基準に直線 / 円弧動作させることができます。

(2) 仕様

<table>
<tr><th>項目</th><th>仕様</th></tr>
<tr><td>対応ロボット</td><td>垂直多関節(6 軸)ロボット、水平多関節(4 軸)ロボット
※ 垂直多関節(5 軸)ロボットとユーザメカでは使用できません。</td></tr>
<tr><td>対応ロボット言語</td><td>MELFA-BASIC V

・MELFA-BASIC V に以下の命令を追加しています。
<table>
<tr><th>命令</th><th>解説</th><th>ページ</th></tr>
<tr><td>EMvs(イームーブエス)</td><td>Ex-T 制御直線補間</td><td>210</td></tr>
<tr><td>EMvc(イームーブシー)</td><td>Ex-T 制御円補間</td><td>204</td></tr>
<tr><td>EMvr(イームーブアール)</td><td>Ex-T 制御円弧補間</td><td>205</td></tr>
<tr><td>EMvr2(イームーブアール 2)</td><td>Ex-T 制御円弧補間 2</td><td>207</td></tr>
<tr><td>EMvr3(イームーブアール 3)</td><td>Ex-T 制御円弧補間 3</td><td>208</td></tr>
</table>
・MELFA-BASIC V の以下の状態変数の機能を変更しています。
<table>
<tr><th>変数名称</th><th>配列要素数</th><th>ページ</th></tr>
<tr><td>P_WkCord</td><td>ワーク座標番号(1 ～ 8)</td><td>365</td></tr>
</table></td></tr>
<tr><td>Ex-T 座標設定個数</td><td>最大 8 個(パラメータ / 状態変数で座標系の設定をおこないます)</td></tr>
</table>

(3) 必要な機器とソフトウェアバージョン

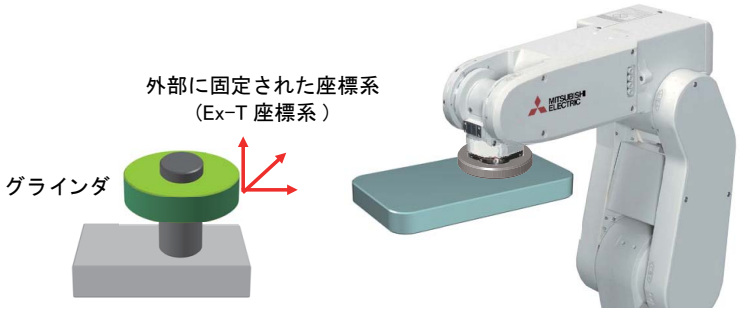
Ex-T 制御は、下表に示すロボットコントローラのソフトウェアバージョンで使用することができます。

表 7-43：Ex-T 制御が使用可能なソフトウェアバージョン

機器	コントローラ	T/B
使用可能なソフトウェアバージョン	F-Q シリーズ：Ver. R5 以降 F-D シリーズ：Ver. S5 以降	R32TB/R33TB：Ver.1.3 以降 R56TB/R57TB：Ver.2.3 以降

7.5.2 Ex-T座標の設定

Ex-T 制御を用いる場合は、外部に固定された基準座標(EX-T 座標)を設定する必要があります。



・固定設置されたグラインダによる研磨作業では、研磨する位置が Ex-T 座標の原点になります。
・固定設置されたディスペンサによる塗布作業では、ノズル先端の位置が Ex-T 座標の原点になります。

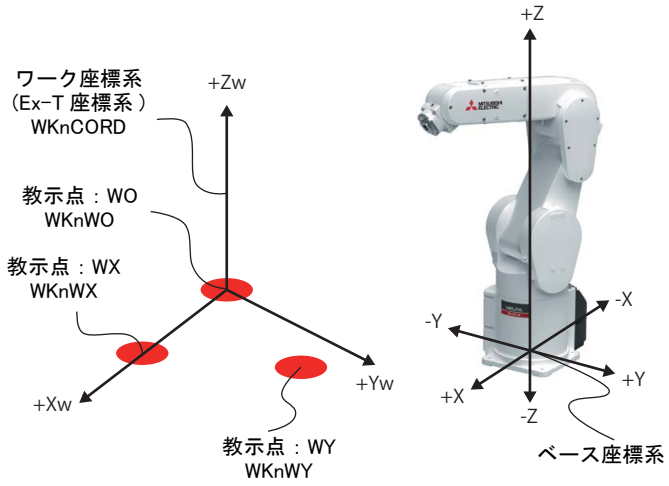
(1) 設定方法

Ex-T 座標の設定は、ワーク座標と同一のパラメータ / 状態変数を用いて設定します。ワーク座標系と同じように

・T/B または RT ToolBox2 を用いてのパラメータ設定
・ロボットプログラム(MELFA-BASIC V)での、システム状態変数への設定

にておこないます。

Ex-T 座標 (ワーク座標) に関するパラメータを下表に示します。

パラメータ名	内容説明
WKnCORD n は 1 ～ 8	ワーク座標(Ex-T座標)の座標値です。 (X, Y, Z, A, B, C) 単位: mm, または度 ワークジョグ操作での基準座標や、ワーク座標データとして使用します。Ex-T制御における制御点(Ex-T座標)としても使用します。 ワーク座標データとして使用する場合、ロボット機種によって有効な軸成分が異なります。434ページの「5.7標準ベース座標について」を参照してください。 T/Bでのワーク座標設定(定義)操作で定義された座標値が設定されますが、本パラメータに座標値を入力することでもワーク座標を定義することができます。この場合，下記の教示によってワーク座標を定義するための3点(パラメータ:WKnWO，WKnWX，WKnWY(nは1～8))の各座標値は0にクリアします。  注)ワーク座標(Ex-T座標)の教示は、ベース変換が初期値の状態(ワールド座標系とベース座標系が一致している状態)でおこなったほうが管理がしやすくなります。特に、ワーク座標を複数定義される場合は混乱防止のために推奨します
WKnWO n は 1 ～ 8	ワーク座標(Ex-T座標)を定義するための教示位置で，ワーク座標(Ex-T座標)の原点の位置です。(T/Bでの教示操作の"WO"に対応します。上図参照) (X, Y, Z) 単位: mm 注)この座標値を入力しただけではワーク座標(Ex-T座標)は定義されません。T/Bワーク座標設定画面の[定義]操作またはRT ToolBox2のワーク座標パラメータ画面の[書き込み]操作でワーク座標の計算処理をおこなってください。
WKnWX n は 1 ～ 8	ワーク座標(Ex-T座標)を定義するための教示位置で，ワーク座標(Ex-T座標)の+X軸上の位置です。(T/Bでの教示操作の"WX"に対応します。上図参照) (X, Y, Z) 単位: mm 注)この座標値を入力しただけではワーク座標(Ex-T座標)は定義されません。T/Bワーク座標設定画面の[定義]操作またはRT ToolBox2のワーク座標パラメータ画面の[書き込み]操作でワーク座標の計算処理をおこなってください。
WKnWY n は 1 ～ 8	ワーク座標(Ex-T座標)を定義するための教示位置で，ワーク座標(Ex-T座標)のX-Y平面上での+Y軸側の位置です。(T/Bでの教示操作の"WY"に対応します。上図参照) (X, Y, Z) 単位: mm 注)この座標値を入力しただけではワーク座標(Ex-T座標)は定義されません。T/Bワーク座標設定画面の[定義]操作またはRT ToolBox2のワーク座標パラメータ画面の[書き込み]操作でワーク座標の計算処理をおこなってください。

WKnWO、WKnWX、WKnWY を用いた 3 点教示でワーク座標 (Ex-T 座標) を設定する場合は、教示の基準位置が制御点になるようにツールデータを設定すると便利です。

R33TB または R32TB を用いて 3 点教示をおこなう操作方法については、別冊の「取扱説明書 / ロボット本体セットアップから保守まで」の "2.4 動作の確認 " の "(6) ワークジョグ操作 " を参照してください。

7.5.3 Ex-Tジョグ

Ex-T ジョグは、ワーク座標 (Ex-T 座標) を制御点としてワーク座標系 (Ex-T 座標系) に沿ってジョグ動作をおこなう機能です。ワーク座標系に沿った動作をおこなうため、従来のワークジョグの動作と似ていますが、Ex-T ジョグと従来のワークジョグでは姿勢成分に関する動作が異なります。

Ex-T ジョグ操作は、TB のワークジョグ操作でおこないます。従来のワークジョグとの動作モードの切替は、それぞれのワーク座標系 (Ex-T 座標系) ごとのパラメータ WK1JOGMD ～ WK8JOGMD の設定にておこないます。

パラメータ名	内容説明
WKnJOGMD n は 1 ～ 8	ワークジョグ操作における動作モードをワーク座標ごとに設定します。 0:ワークジョグ(A,B,Cの動作は、ワーク座標のX軸、Y軸、Z軸と平行な軸回りに回転します。制御点の位置は変わりません。) 1:Ex-Tジョグ(A,B,Cの動作は、ワーク座標のX軸、Y軸、Z軸を中心として制御点を移動しながら回転します。)

パラメータ WK1JOGMD ～ WK8JOGMD の設定による動作の違いを、ロボット機種ごとに表に示します。

RV6 軸タイプのワークジョグの動作

ワークジョグの動作モード	ワークジョグ	Ex-T ジョグ
パラメータ WKnJOGMD(n=1 ～ 8) の設定	0(初期値)	1
XYZ キーの動作	ワーク座標系の各軸に沿って移動します。	従来のワークジョグと同じです。
ABC キーの動作	制御点の位置を保ったまま、ワーク座標系に沿って、向きを変えます。	ワーク座標系の各軸を中心軸として、制御点の位置を変えながら、向きを変えます。

RH4 軸タイプ、RH4 軸天吊りタイプ

ワークジョグの動作モード	ワークジョグ	Ex-T ジョグ
パラメータ WKnJOGMD(n=1 ～ 8) の設定	0(初期値)	1
XYZ キーの動作	ワーク座標系の各軸に沿って移動します	従来のワークジョグと同じです
C キーの動作	制御点の位置を保ったまま、ワーク座標系に沿って、向きを変えます。	ワーク座標系 (EX-T 座標系) の Z 軸 (Zw) を中心軸として、制御点の位置を変えながら、向きを変えます。
AB キーの動作	ロボットは動作しません。	ロボットは動作しません。

従来のワークジョグと Ex-T ジョグでは、ロボットの姿勢成分の動作が異なります。ここでは、C 成分の動作を例として違いを説明します。

(1) ワークジョグにおける姿勢成分動作

ワークジョグにおける姿勢成分のジョグ動作は、制御点においてワーク座標の XYZ 軸に平行な軸周りの回転となります。このとき、位置は固定されたままとなります。

ワークジョグにおける C 成分動作例を図 7-120 に示します。

W0-Wx-Wy はワーク座標系を示します(ワーク座標系を +Wz から見た図としています)。●はロボットの制御点 (TCP)、角丸の四角はロボットが把持しているワークです。点線は移動後のワーク位置(姿勢)を示します。

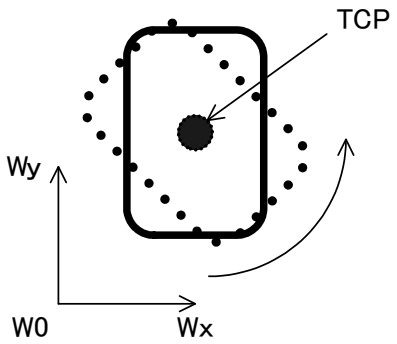


図 7-120：ワークジョグの C 成分動作

(2) Ex-T ジョグにおける姿勢成分動作

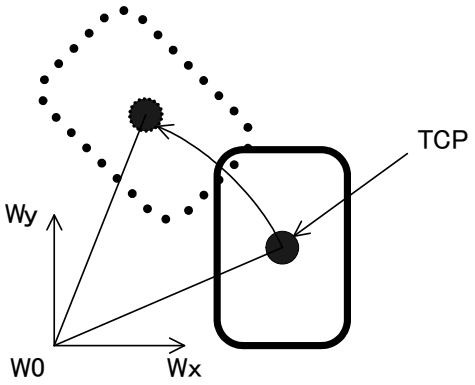
Ex-T ジョグにおける姿勢成分のジョグ動作は、Ex-T 座標系((ワーク座標系)の XYZ 軸周りの回転となります。このとき、ロボットの位置も変化します。

Ex-T ジョグにおける C 成分動作例を図 7-121 の＜動作例 1>、＜動作例 2> に示します。

W0-Wx-Wy は Ex-T 座標系 (ワーク座標系) を示します (Ex-T 座標系を +Wz から見た図としています)。●はロボットの制御点 (TCP)、角丸の四角はロボットが把持しているワークです。点線は移動後のワーク位置 (姿勢) を示します。

< 動作例 1> は Ex-T 座標系 (ワーク座標系) とワークが離れている場合、< 動作例 2> は Ex-T 座標系 (ワーク座標系) の原点 (または Z 軸) がワークと接している場合です。ともに W0 を中心に回転動作します。

< 動作例 1>



< 動作例 2>

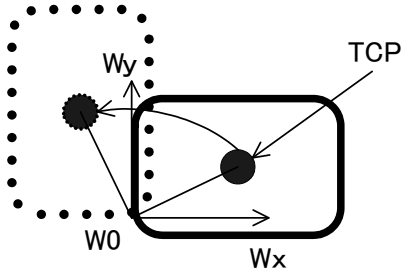


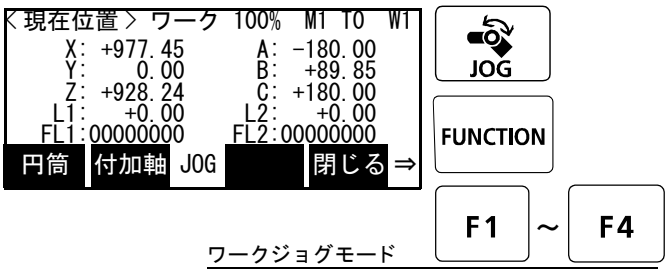
図 7-121 : Ex-T ジョグの C 成分動作

(3) Ex-T ジョグの操作

Ex-T ジョグの操作は、ワークジョグの操作と同じ方法です。

EX-T 座標 (ワーク座標) と、動作モード (パラメータ WK1JOGMD ～ WK8JOGMD) を予め設定しておく必要があります。

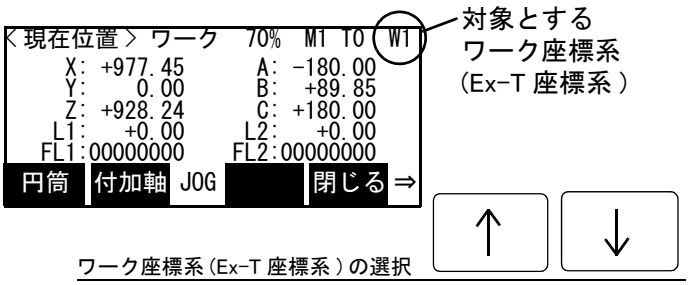
ワークジョグモードの選択



注) 表示される座標値は直交座標系に基づいた値です。

[JOG] キーを押してジョグ画面を表示させます。(画面下側に ″JOG″ が表示されています) 画面の上にジョグモードの ″ ワーク ″ が表示されていることを確認します。
他のジョグモードが表示されている場合は、″ ワーク ″ に該当するファンクションキーを押してください。
(画面下に希望するジョグモードが表示されていない場合は、[FUNCTION] キーを押すと表示されます)
ジョグ操作を終了する場合は、[JOG] キーを再度押すか、″ 閉じる ″ に該当するファンクションキーを押してください。

ワーク座標系 (Ex-T 座標系) の確認・選択



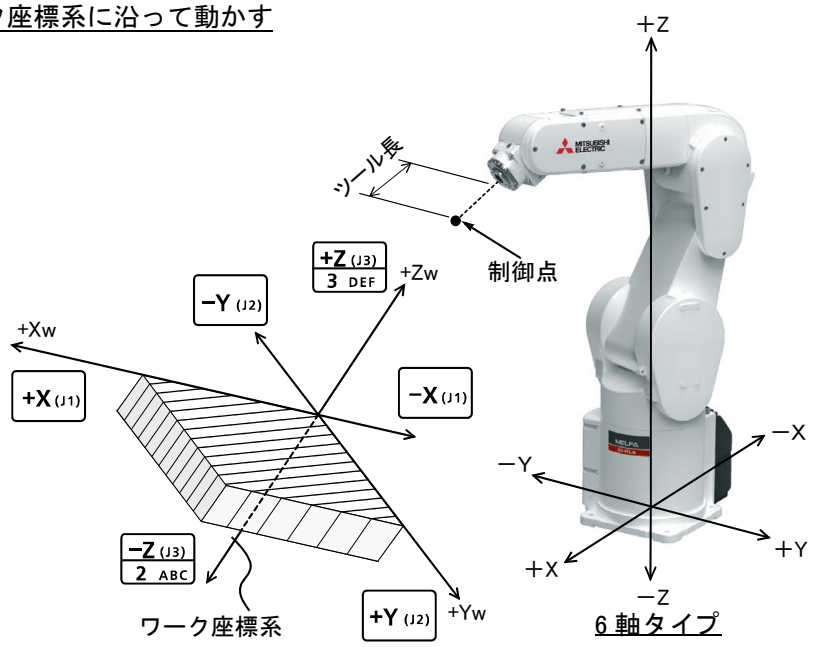
ワークジョグ (Ex-T ジョグ) 動作の対象とするワーク座標系 (Ex-T 座標系) を確認してください。画面右上に 8 個のワーク座標系 (Ex-T 座標系) の内、現在対象となっている番号を表示しています。(W1 ～ W8)
希望するワーク座標系 (Ex-T 座標系) が表示されていない場合は矢印キー ([↑], [↓]) で変更できます。
[↑] キーを押す毎に W1 → W2 ···· W7 → W8 の順に上がり、[↓] キーを押すと逆に下がります。

⚠注意 対象とするワーク座標系 (Ex-T 座標系) の番号が、正しく表示されていることを必ず確認してください。(画面右上の W1 ～ W8 の表示)
間違ったワーク座標系 (Ex-T 座標系) では、ロボットは意図せぬ方向へ動作するので物損や人身事故の原因となります。

(4) RV6 軸タイプのワークジョグ動作

XYZ キーでの動作は、ワークジョグモードと Ex-T ジョグモードでは、同じ動作をします。

ワーク座標系に沿って動かす



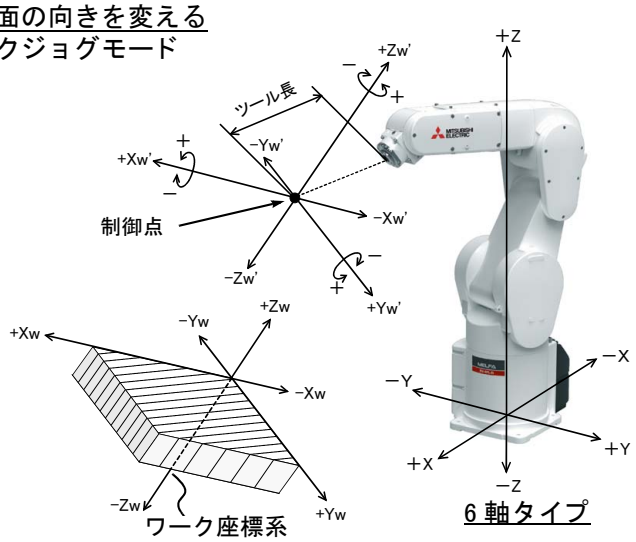
※ フランジの向きは変わりません。
ワーク座標系に沿って制御点を直線で移動させます。

・[+X(J1)] キーを押すとワーク座標系の X 軸 (Xw) のプラス方向に沿って移動します。
[−X(J1)] キーを押すとマイナス方向に沿って移動します。
・[+Y(J2)] キーを押すとワーク座標系の Y 軸 (Yw) のプラス方向に沿って移動します。
[−Y(J2)] キーを押すとマイナス方向に沿って移動します。
・[+Z(J3)] キーを押すとワーク座標系の Z 軸 (Zw) のプラス方向に沿って移動します。
[−Z(J3)] キーを押すとマイナス方向に沿って移動します。

ABC キーの動作は、ワークジョグモードと Ex-T ジョグモードでは、異なった動作をします。

フランジ面の向きを変える

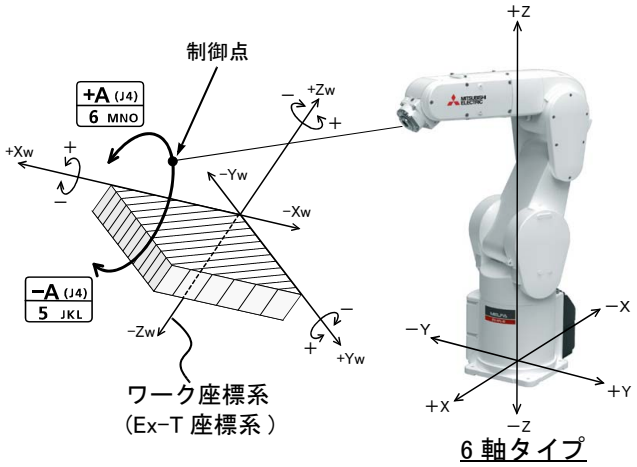
①ワークジョグモード



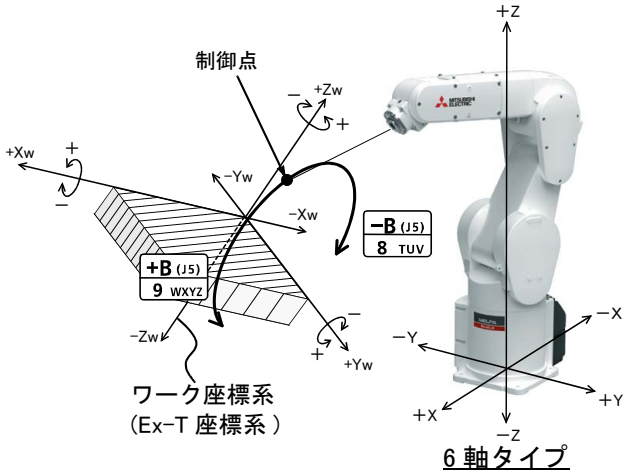
※ フランジの位置は変わりません。
ワーク座標系に沿ってフランジの向きを変えます。

・[+A(J4)] キーを押すと X 軸のプラス回りに回転します。
[−A(J4)] キーを押すとマイナス回りに回転します。
・[+B(J5)] キーを押すと Y 軸のプラス回りに回転します。
[−B(J5)] キーを押すとマイナス回りに回転します。
・[+C(J6)] キーを押すと Z 軸のプラス回りに回転します。
[−C(J6)] キーを押すとマイナス回りに回転します。

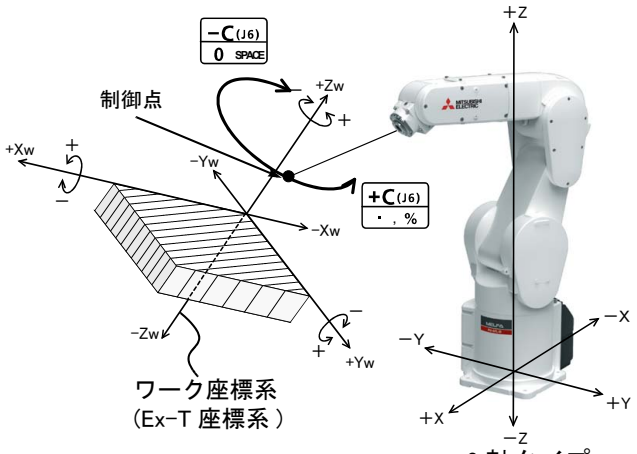
② Ex-T ジョグモード



6 軸タイプ



6 軸タイプ



6 軸タイプ

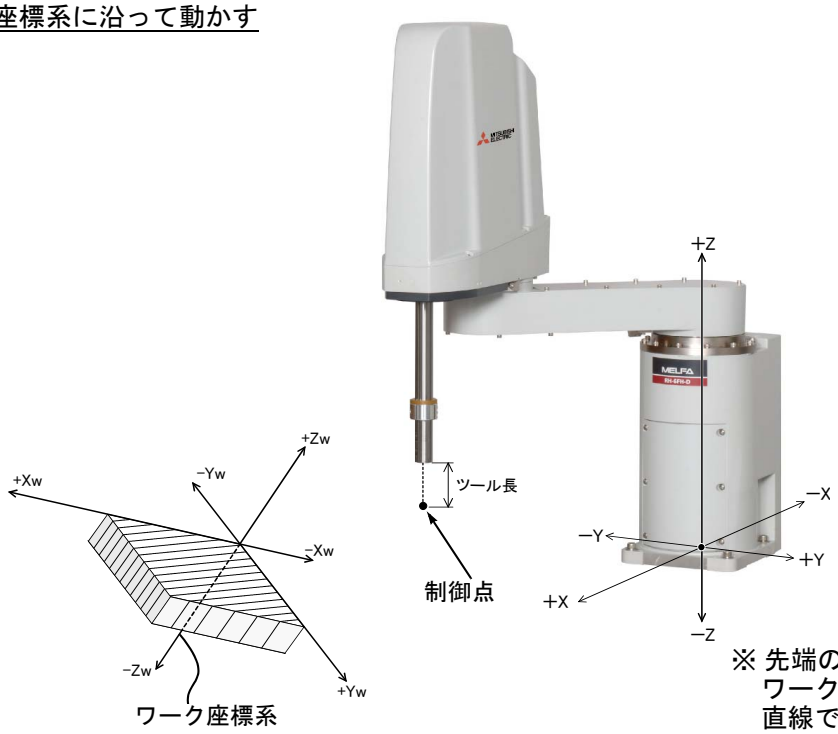
※ ワーク座標系 (Ex-T 座標系) の各軸を中心に、制御点が回転します。
[+A(J4)] キーと [-A(J4)] キーでは Xw 軸を中心に、[+B(J5)] キーと [-B(J5)] キーでは Yw 軸を中心に、
[+C(J6)] キーと [-C(J6)] キーでは Zw 軸を中心に制御点が回転するような動作をおこないます。

・[+A(J4)] キーを押すとワーク座標系 (Ex-T 座標系) の X 軸 (Xw) を中心としてプラス回りに制御点が回転します。
[-A(J4)] キーを押すとマイナス回りに回転します。
・[+B(J5)] キーを押すとワーク座標系 (Ex-T 座標系) の Y 軸 (Yw) を中心としてプラス回りに制御点が回転します。
[-B(J5)] キーを押すとマイナス回りに回転します。
・[+C(J6)] キーを押すとワーク座標系 (Ex-T 座標系) の Z 軸 (Zw) を中心としてプラス回りに制御点が回転します。
[-C(J6)] キーを押すとマイナス回りに回転します。

(5) RH4 軸タイプのワークジョグ動作

XYZ キーでの動作は、ワークジョグモードと Ex-T ジョグモードでは、同じ動作をします。

ワーク座標系に沿って動かす

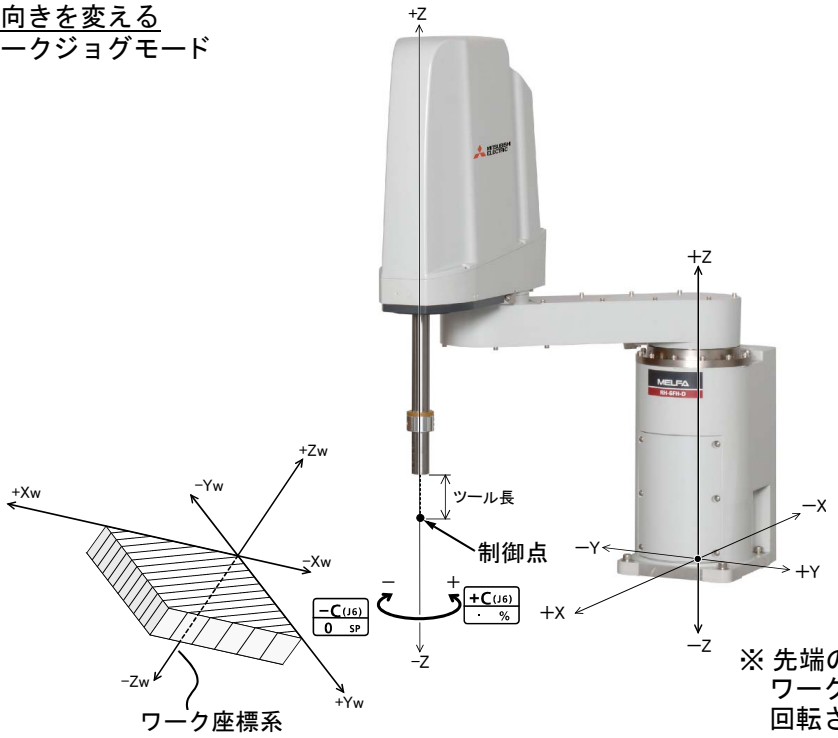


※ 先端の向きは変わりません。
ワーク座標系に沿って制御点を直線で移動させます。

・[+X(J1)] キーを押すとワーク座標系の X 軸 (Xw) のプラス方向に沿って移動します。
[−X(J1)] キーを押すとマイナス方向に沿って移動します。
・[+Y(J2)] キーを押すとワーク座標系の Y 軸 (Yw) のプラス方向に沿って移動します。
[−Y(J2)] キーを押すとマイナス方向に沿って移動します。
・[+Z(J3)] キーを押すとワーク座標系の Z 軸 (Zw) のプラス方向に沿って移動します。
[−Z(J3)] キーを押すとマイナス方向に沿って移動します。

C キーの動作は、ワークジョグモードと Ex-T ジョグモードでは、異なった動作をします。
AB キーは動作しません。

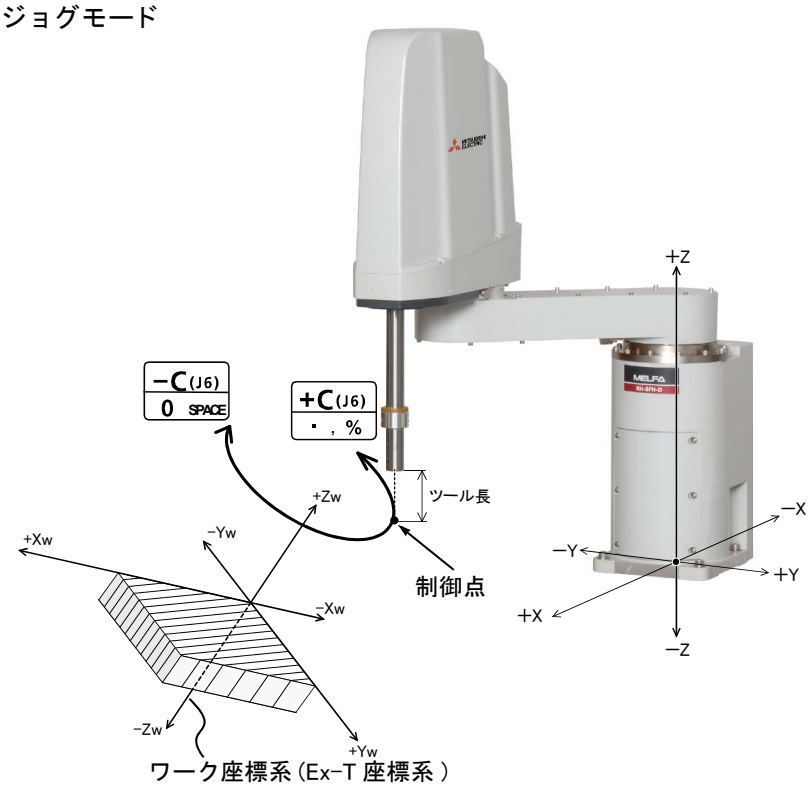
先端の向きを変える
①ワークジョグモード



※ 先端の位置は変わりません。
ワーク座標系に沿って先端軸を回転させます。

・[+C(J6)] キーを押すと先端がプラス方向に回転します。
[-C(J6)] キーを押すとマイナス方向に回転します。

② Ex-T ジョグモード

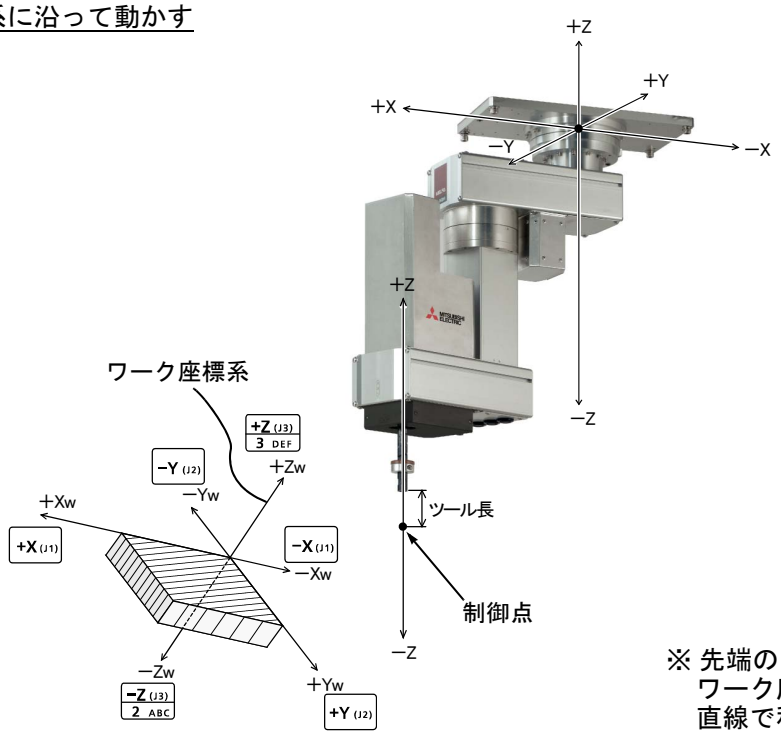


・[+C(J6)] キーを押すとワーク座標系 (Ex-T 座標系) の Z 軸 (Zw) を中心としてプラス回りに制御点が回転します。
[-C(J6)] キーを押すとマイナス回りに回転します。

(6) RH4 軸天吊りタイプのワークジョグ動作

XYZ キーでの動作は、ワークジョグモードと Ex-T ジョグモードでは、同じ動作をします。

ワーク座標系に沿って動かす

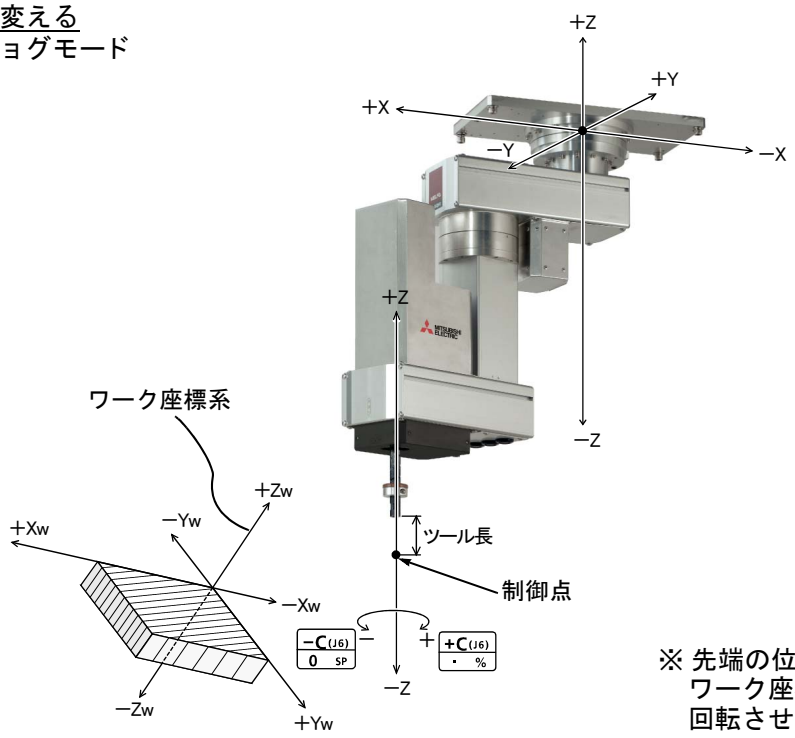


※ 先端の向きは変わりません。
ワーク座標系に沿って制御点を直線で移動させます。

・[+X(J1)] キーを押すとワーク座標系の X 軸 (Xw) のプラス方向に沿って移動します。
[−X(J1)] キーを押すとマイナス方向に沿って移動します。
・[+Y(J2)] キーを押すとワーク座標系の Y 軸 (Yw) のプラス方向に沿って移動します。
[−Y(J2)] キーを押すとマイナス方向に沿って移動します。
・[+Z(J3)] キーを押すとワーク座標系の Z 軸 (Zw) のプラス方向に沿って移動します。
[−Z(J3)] キーを押すとマイナス方向に沿って移動します。

C キーの動作は、ワークジョグモードと Ex-T ジョグモードでは、異なった動作をします。
AB キーは動作しません。

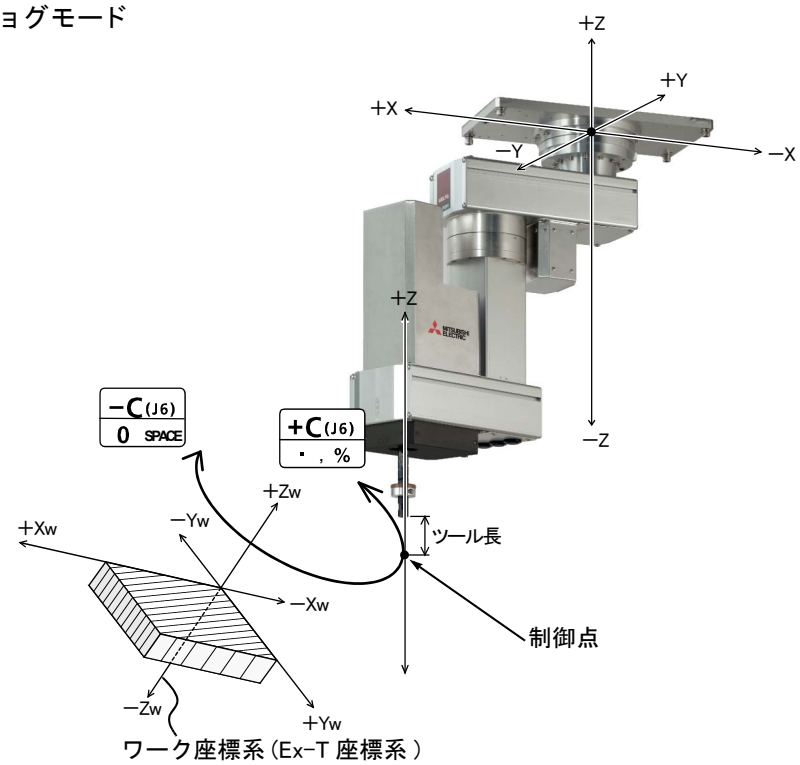
先端の向きを変える
①ワークジョグモード



※ 先端の位置は変わりません。
ワーク座標系に沿って先端軸を回転させます。

・[+C(J6)] キーを押すと先端がプラス方向に回転します。
[-C(J6)] キーを押すとマイナス方向に回転します。

② Ex-T ジョグモード



・[+C(J6)] キーを押すとワーク座標系 (Ex-T 座標系) の Z 軸 (Zw) を中心としてプラス回りに制御点が回転します。
[-C(J6)] キーを押すとマイナス回りに回転します。

7.5.4 ロボットプログラムの作成

(1) Ex-T 制御関連命令・変数の一覧

Ex-T 制御に関わる MELFA-BASIC V の命令と変数の一覧を示します。
命令の詳細説明は、表の参照ページを参照してください。

表 7-44：Ex-T 制御関連の命令

No.	命令の種類	追加命令	機能	参照ページ
1	直線補間	EMvs（イームーブエス）	ワーク座標（Ex-T 座標）に沿った直線補間	210
2	円補間	EMvc（イームーブシー）	ワーク座標（Ex-T 座標）に沿った円補間	204
3	円弧補間	EMvr（イームーブアール）	ワーク座標（Ex-T 座標）に沿った円弧補間	205
4		EMvr2（イームーブアール 2）		207
5		EMvr3（イームーブアール 3）		208

表 7-45：Ex-T 制御関連の状態変数

No.	変数名称	機能概要	補足説明	参照ページ
1	P_WkCord	ワーク座標の読出し、設定	Ex-T 座標も兼用します。	365

(2) プログラミング例

図のような動作をさせるためのプログラム例を示します。
固定された加工治具に、ロボットが把持したワークをなぞらせる動作をおこないます。（図①～⑤）

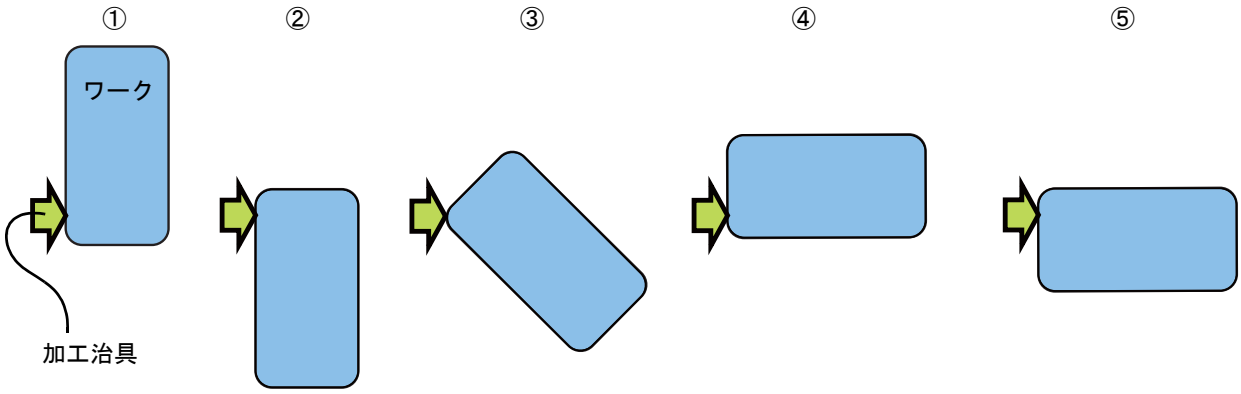


図 7-122：作業例 1

この図を重ね合わせて実際の動作を示すと、次のようになります。

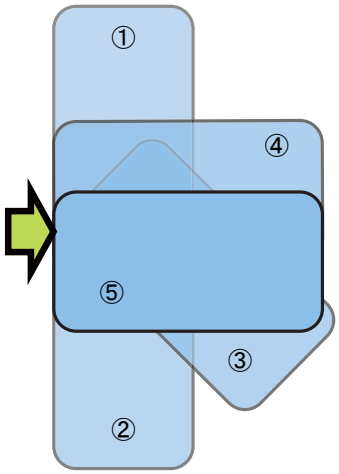
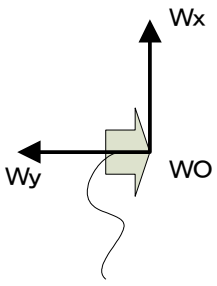


図 7-123：作業例 2

■手順 1：ワーク座標(Ex-T 座標)の設定
　図の加工治具のワークとの接触部位が、ワーク座標(Ex-T 座標)原点になるようにワーク座標(Ex-T 座標)を設定します。(ここでは、ワーク座標 1 に設定したとします。)



　また、このワーク座標系に沿ったジョグ動作をおこなうことができるようにするために、パラメータ WK1JOGMD を「1(Ex-T ジョグモード)」に設定しておきます。

■手順 2：位置の教示
　実際にロボットにワークを把持させて、位置の教示をおこないます。
ここでは、図の①～⑤の位置を教示します。
位置教示をおこなうときに、ワークジョグ(Ex-T ジョグ)にてロボットのジョグ送りをおこなうと、加工治具に沿ったジョグ送りをおこなうことができます。

■手順 3：プログラムの作成
　MELFA-BASIC V のプログラムを作成します。
　(実際には、ワークの把持動作や信号入出力などが必要になりますが省略しています。)

【プログラム例】
```
Mov P001                ' ①の位置へ移動
Dly 0.5
Spd 50                  ' 加工速度(ワークの移動速度)を 50mm/sec に設定
EMvs 1, P002            ' ②の位置へワーク座標 1 に沿って Ex-T 直線補間で移動
EMvr 1, P002,P003,P004  ' ②から③を経由して④へワーク座標 1 に沿って Ex-T 円弧補間で移動。
EMvs 1, P005            ' ⑤の位置へワーク座標 1 に沿って Ex-T 直線補間で移動
  :
```

　この例では、教示位置を 5 点としていますが、実際にはワーク形状や加工の具合により、教示位置を増やす必要があります。それに合わせて、プログラムも変更します。

7.6 技術相談窓口のお知らせ

本書では、お客様がロボットの取扱、操作やプログラミングをおこなうことを想定して、できるだけわかりやすく説明しておりますが、お読みいただいても分かりにくいことなどの相談窓口として、「MELFA テレホンセンター」を開設いたしております。どうぞお気軽にご相談ください。

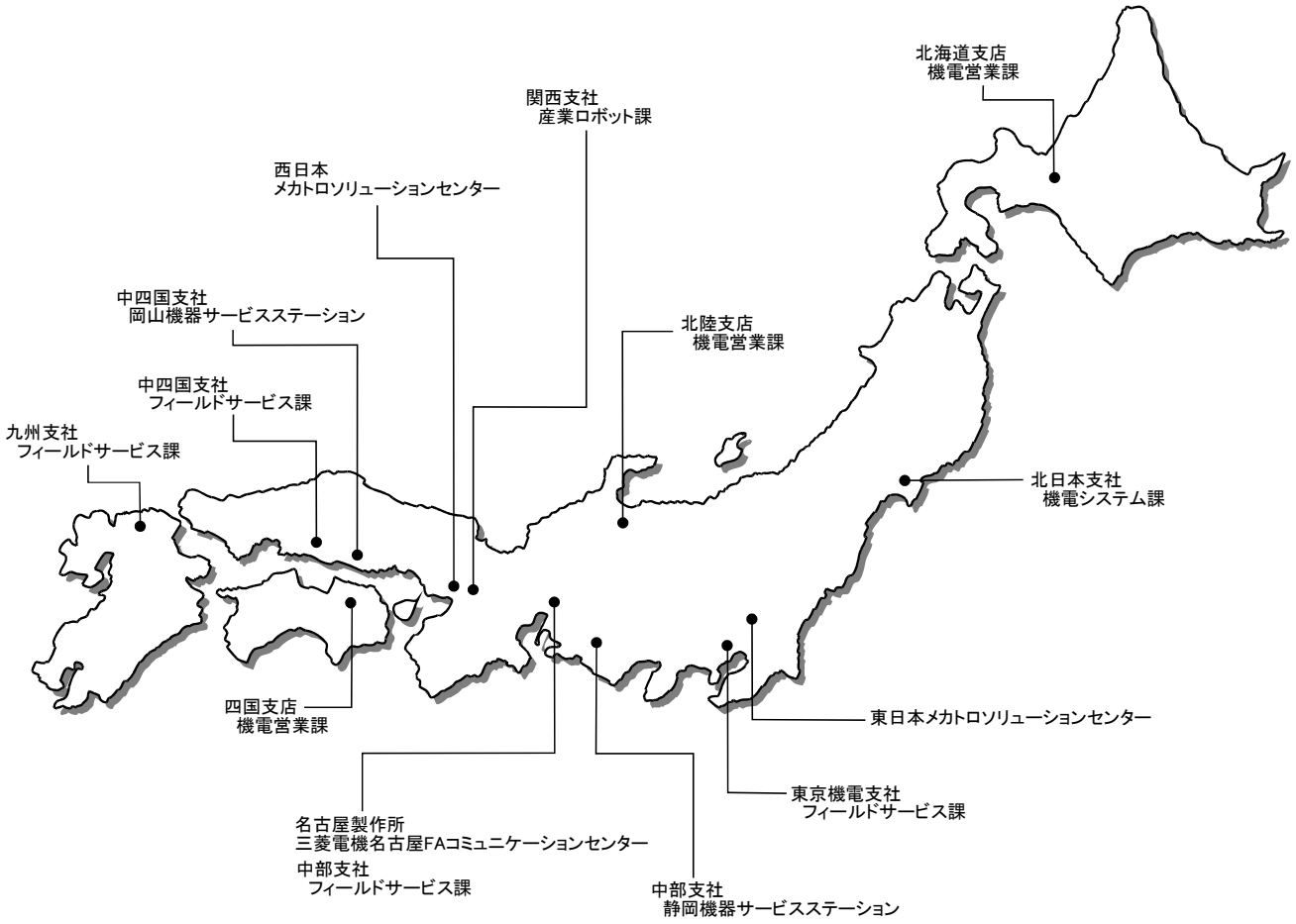
〈MELFA テレホンセンター〉

相談内容　：ロボットの仕様、機能、および導入後の立上、取扱、運転、
　　操作、プログラミング等についての技術相談を承ります。
電話番号　：052-721-0100(直通)
開設時間　：月曜日～金曜日.....9：00 ～ 19：00
　　土・日・祝日.....9：00 ～ 17：00

7.7 アフターサービスについて

ロボットの修理、点検などの保守サービスについては、裏面の三菱電機システムサービス (株) が窓口となります。ご用の際は最寄の三菱電機システムサービス (株) までご連絡ください。

三菱電機産業用ロボット保守サービスネットワーク



三菱電機システムサービス株式会社

お問い合わせは下記へどうぞ

北日本支社
　機電システム課----------〒983-0013 仙台市宮城野区中野1-5-35 ----------(022)353-7814
北海道支店
　機電営業課 ----------〒004-0041 札幌市厚別区大谷地東2-1-18 ----------(011)890-7515
東京機電支社
　フィールドサービス課 ----------〒108-0022 東京都港区海岸3-19-22(三菱倉庫芝浦ビル) -------(03)3454-2561
中部支社
　フィールドサービス課 ----------〒461-8675 名古屋市東区矢田南5-1-14 ----------(052)722-7601
　静岡機器サービスステーション ----------〒422-8058 静岡市駿河区中原877-2 ----------(054)287-8866
北陸支店
　機電営業課 ----------〒920-0811 金沢市小坂町北255----------(076)252-9519
関西支社
　産業ロボット課----------〒531-0076 大阪市北区大淀中1-4-13 ----------(06)6454-0191
中四国支社
　フィールドサービス課 ----------〒732-0802 広島市南区大州4-3-26----------(082)285-2111
　岡山機器サービスステーション ----------〒700-0951 岡山市北区田中606-8----------(086)242-1900
四国支店
　機電営業課 ----------〒760-0072 高松市花園町1-9-38----------(087)831-3186
九州支社
　フィールドサービス課 ----------〒812-0007 福岡市博多区東比恵3-12-16 ----------(092)483-8208

営業体制
　9:00～19:00の間は、全国の支社・支店・サービスステーションでお受けいたします。
平日夜間および土・日・祝日受付体制
　平日の19:00～翌朝9:00、および土・日・祝日の9:00～翌朝9:00は、集中受付センター(03)5460-3582にてお受けいたします。
　夜間および休日につきましては、集中センターで受け付け、修理に関するご相談には技術者が待機してバックアップいたします。

三菱電機株式会社
　名古屋製作所 ----------〒461-8670 名古屋市東区矢田南5-1-14 ----------(052)712-2609

三菱電機・産業用ロボット

三菱電機株式会社 〒100-8310　東京都千代田区丸の内 2-7-3(東京ビル)
お問い合わせは下記へどうぞ
本社 ..... 〒 100-8310 東京都千代田区丸の内 2-7-3 (東京ビル) .....(03)3218-6740
北海道支社 ..... 〒 060-0002 札幌市中央区北 2 条西 4-1 (北海道ビル) .....(011)212-3794
東北支社 ..... 〒 980-0011 仙台市青葉区上杉 1-17-7 (仙台上杉ビル) .....(022)216-4546
北陸支社 (金沢) ..... 〒 920-0031 金沢市広岡 3-1-1 (金沢パークビル 4F) .....(076)233-5502
中部支社 ..... 〒 451-8522 名古屋市西区牛島町 6-1 (名古屋ルーセントタワー) .....(052)565-3326
関西支社 ..... 〒 530-8206 大阪市北区大深町 4-20 (グランフロント大阪　タワー A) .....(06)6486-4120
中国支社 ..... 〒 730-8657 広島市中区中町 7-32 (ニッセイ広島ビル) .....(082)248-5445
四国支社 ..... 〒 760-8654 高松市寿町 1-1-8 (日本生命高松駅前ビル) .....(087)825-0055
九州支社 ..... 〒 810-8686 福岡市中央区天神 2-12-1 (天神ビル) .....(092)721-2251

この印刷物は2015年9月の発行です。なお、お断りなしに仕様を変更することがありますのでご了承ください。 2015年9月作成　(1509)MEE